

sensors

Advances in Multi-Sensor Information Fusion: Theory and Applications 2017

Edited by
Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang
Printed Edition of the Special Issue Published in *Sensors*

Advances in Multi-Sensor Information Fusion: Theory and Applications 2017

Special Issue Editors

Xue-Bo Jin

Shuli Sun

Hong Wei

Feng-Bao Yang

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade



Special Issue Editors

Xue-Bo Jin
Beijing Technology and Business University
China

Shuli Sun
Heilongjiang University
China

HongWei
University of Reading
UK

Feng-Bao Yang
North University of China
China

Editorial Office

MDPI
St. Alban-Anlage 66
Basel, Switzerland

This edition is a reprint of the Special Issue published online in the open access journal *Sensors* (ISSN 1424-8220) from 2017–2018 (available at: http://www.mdpi.com/journal/sensors/special_issues/theory_applications_2017).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, F.M.; Lastname, F.M. Article title. <i>Journal Name</i> Year , Article number, page range.

First Edition 2018

ISBN 978-3-03842-933-3 (Pbk)

ISBN 978-3-03842-934-0 (PDF)

Cover photo courtesy of Xue-Bo Jin

Articles in this volume are Open Access and distributed under the Creative Commons Attribution license (CC BY), which allows users to download, copy and build upon published articles even for commercial purposes, as long as the author and publisher are properly credited, which ensures maximum dissemination and a wider impact of our publications. The book taken as a whole is © 2018 MDPI, Basel, Switzerland, distributed under the terms and conditions of the Creative Commons license CC BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Table of Contents

About the Special Issue Editors	vii
Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang Advances in Multi-Sensor Information Fusion: Theory and Applications 2017 doi: 10.3390/s18041162	1
Yanpeng Liu, Yibin Li, Xin Ma and Rui Song Facial Expression Recognition with Fusion Features Extracted from Salient Facial Areas doi: 10.3390/s17040712	3
Wenhua Guo, Zuren Feng and Xiaodong Ren Object Tracking Using Local Multiple Features and a Posterior Probability Measure doi: 10.3390/s17040739	21
Hua Liu and Wen Wu Strong Tracking Spherical Simplex-Radial Cubature Kalman Filter for Maneuvering Target Tracking doi: 10.3390/s17040741	38
Biao Hu, Guang Chen, Uzair Sharif, Rajat Koner, Kai Huang, Walter Stechele and Alois Knoll Random Finite Set Based Bayesian Filtering with OpenCL in a Heterogeneous Platform doi: 10.3390/s17040843	51
Xiaobin Xu, Zhenghui Li, Guo Li and Zhe Zhou State Estimation Using Dependent Evidence Fusion: Application to Acoustic Resonance- Based Liquid Level Measurement doi: 10.3390/s17040924	70
Liang-Qun Li, Xiao-Li Wang, Zong-Xiang Liu and Wei-Xin Xie Auxiliary Truncated Unscented Kalman Filtering for Bearings-Only Maneuvering Target Tracking doi: 10.3390/s17050972	89
Quanbo Ge, Zhongliang Wei, Tianfa Cheng, Shaodong Chen and Xiangfeng Wang Flexible Fusion Structure-Based Performance Optimization Learning for Multisensor Target Tracking doi: 10.3390/s17051045	103
Ling Bao Kong, Ming Jun Ren and Min Xu Development of Data Registration and Fusion Methods for Measurement of Ultra-Precision Freeform Surfaces doi: 10.3390/s17051110	119
Raquel Caballero-Águila, Aurora Hermoso-Carazo and Josefa Linares Pérez Optimal Fusion Estimation with Multi-Step Random Delays and Losses in Transmission doi: 10.3390/s17051151	134
Baoyu Liu, Xingqun Zhan and Zheng H. Zhu Multisensor Parallel Largest Ellipsoid Distributed Data Fusion with Unknown Cross-Covariances doi: 10.3390/s17071526	151

Wen Jiang, Miaoyan Zhuang and Chunhe Xie A Reliability-Based Method to Sensor Data Fusion doi: 10.3390/s17071575	171
Sheng-lun Yi, Xue-bo Jin, Ting-li Su, Zhen-yun Tang, Fa-fa Wang, Na Xiang and Jian-lei Kong Online Denoising Based on the Second-Order Adaptive Statistics Model doi: 10.3390/s17071668	191
Wen Jiang, Ying Cao, Lin Yang and Zichang He A Time-Space Domain Information Fusion Method for Specific Emitter Identification Based on Dempster–Shafer Evidence Theory doi: 10.3390/s17091972	208
Xinyang Deng and Wen Jiang Fuzzy Risk Evaluation in Failure Mode and Effects Analysis Using a D Numbers Based Multi-Sensor Information Fusion Method doi: 10.3390/s17092086	227
Jingbin Wang, Xiaohong Wang and Lizhi Wang Modeling of BN Lifetime Prediction of a System Based on Integrated Multi-Level Information doi: 10.3390/s17092123	244
Xiaoli Meng, Heng Wang and Bingbing Liu A Robust Vehicle Localization Approach Based on GNSS/IMU/DMI/LiDAR Sensor Fusion for Autonomous Vehicles doi: 10.3390/s17092140	263
Yongchuan Tang, Deyun Zhou, Miaoyan Zhuang, Xueyi Fang and Chunhe Xie An Improved Evidential-IOWA Sensor Data Fusion Approach in Fault Diagnosis doi: 10.3390/s17092143	282
Ángel J. Jarama, Jaime López-Araquistain, Gonzalo de Miguel and Juan A. Besada Complete Systematic Error Model of SSR for Sensor Registration in ATC Surveillance Networks doi: 10.3390/s17102171	297
Lifan Sun, Baofeng Ji, Jian Lan, Zishu He and Jiexin Pu Tracking of Maneuvering Complex Extended Object with Coupled Motion Kinematics and Extension Dynamics Using Range Extent Measurements doi: 10.3390/s17102184	325
Shaoyi Liang, Deqiang Han A Clustering-Oriented Closeness Measure Based on Neighborhood Chain and Its Application in the Clustering Ensemble Framework Based on the Fusion of Different Closeness Measures doi: 10.3390/s17102226	344
Yun Li, Shu Li Sun and Gang Hao A Weighted Measurement Fusion Particle Filter for Nonlinear Multisensory Systems Based on Gauss–Hermite Approximation doi: 10.3390/s17102222	372
Lichuan Zhang, Tonghao Wang, Feihu Zhang and Demin Xu Cooperative Localization for Multi-AUVs Based on GM-PHD Filters and Information Entropy Theory doi: 10.3390/s17102286	386

Fengjian Shi, Xiaoyan Su, Hong Qian, Ning Yang and Wenhua Han	
Research on the Fusion of Dependent Evidence Based on Rank Correlation Coefficient doi: 10.3390/s17102362	402
Fuyuan Xiao	
A Novel Evidence Theory and Fuzzy Preference Approach-Based Multi-Sensor Data Fusion Technique for Fault Diagnosis doi: 10.3390/s17112504	417
Yu Liu, Jun Liu, Gang Li, Lin Qi, Yaowen Li and You He	
Centralized Multi-Sensor Square Root Cubature Joint Probabilistic Data Association doi: 10.3390/s17112546	437
Yanran Wang, Gang Xiao and Zhouyun Dai	
Integrated Display and Simulation for Automatic Dependent Surveillance–Broadcast and Traffic Collision Avoidance System Data Fusion doi: 10.3390/s17112611	453
Babak Khodabandeloo, Dyan Melvin and Hongki Jo	
Model-Based Heterogeneous Data Fusion for Reliable Force Estimation in Dynamic Structures under Uncertainties doi: 10.3390/s17112656	476
Omid Dehzangi, Mojtaba Taherisadr, Raghuvendar Chandalvala	
IMU-Based Gait Recognition Using Convolutional Neural Networks and Multi-Sensor Fusion doi: 10.3390/s17122735	500
Chaoyang Shi, Bi Yu Chen, William H. K. Lam and Qingquan Li	
Heterogeneous Data Fusion Method to Estimate Travel Time Distributions in Congested Road Networks doi: 10.3390/s17122822	522
Christophe Boucher and Jean-Charles Noyer	
A General Framework for 3-D Parameters Estimation of Roads Using GPS, OSM and DEM Data doi: 10.3390/s18010041	543

About the Special Issue Editors

Xue-Bo Jin was born in Liaoning in 1972, China. She received a B.E. degree in industrial electrical and automation and a Master degree in industrial automation from Jilin University, Jilin, China, in 1994 and 1997, and a Ph.D. degree in control theory and control engineering from Zhejiang University, Zhejiang, China, in 2004. From 1997 to 2012 she was working for the College of Informatics and Electronics, Zhejiang Sci-Tech University. Since 2012 she has been a Professor at the College of Computer and Information Engineering, Beijing Technology and Business University. Her research interests include multisensor fusion, statistical signal processing, video/image processing, robust filtering, and Bayesian theory. In particular, her present major interest is multisensor fusion, Bayesian estimation and big data tendency analyses.

Shuli Sun was born in Heilongjiang Province, China in 1971. He received his B.S. degree in mathematics and an M.E. degree in automation from Heilongjiang University, China, in 1996 and 1999, respectively. He earned his Ph.D. at the Harbin Institute of Technology, China, in 2004. He was a Research Fellow in Nanyang Technological University, Singapore from April 2006 to October 2007. Since 2006, he has been a Professor at the Heilongjiang University, China. His main research interests are state estimations, signal processing, information fusion and sensor networks.

Hong Wei obtained her Ph.D degree in Birmingham University in 1996. She joined the University of Reading in October 2000. She has had 18 year post-doctoral research experience in the area of computer vision and image processing. She designed a novel method for measuring the spatial resolution of CMOS cameras, and developed digital image processing algorithms to improve the spatial resolution by using multiple low-resolution images. The research was based on modern image processing technology. Dr. Wei is an expert in the field of image analysis and pattern classification. Her research achievements has secured funding applications to EPSRC, DTI, ESRC, and the Royal Society for various research projects. Her current research interest is intelligent computer vision and multisensor data fusion, and its application in remotely sensed images and in face recognition (biometric).

Feng-Bao Yang was born in Shanxi in 1968, China. He received his B.S degree in Physics in 1991, a Master degree in Systems Engineering in 1997, and a Ph.D. degree in test and measurement technology and instrumentation in 2003 from North University of China, Shanxi, China. From 2004 to 2006, he did Postdoctoral research in Optic Engineering at the Beijing Institute of Technology. Then he spent two years as a senior visiting scholar in Stony Brook University and University of Reading. Dr. Yang is a professor in the School of Information and Communication Engineering, North University of China. His research interests include infrared image processing, multi-source and heterogeneous image registration, recognition and fusion, multi-sensor uncertain information processing, possibility theory and evidence theory. His research achievements have secured funding applications to NSFC and Natural Science Foundation of Shanxi Province.

Editorial

Advances in Multi-Sensor Information Fusion: Theory and Applications 2017

Xue-Bo Jin ^{1,*}, Shuli Sun ², Hong Wei ³ and Feng-Bao Yang ⁴

¹ School of Computer Information and Engineering, Beijing Technology and Business University, Beijing 100048, China

² School of Electronics Engineering, Heilongjiang University, Harbin 150080, China; sunsl@hlju.edu.cn

³ Department of computer science, University of Reading, RG6 6AY Reading, UK; h.wei@reading.ac.uk

⁴ School of Information and Communication Engineering, North University of China, Taiyuan 030051, China; yangfb@nuc.edu.cn

* Correspondence: jinxuebo@btbu.edu.cn

Received: 8 April 2018; Accepted: 9 April 2018; Published: 11 April 2018

Abstract: The information fusion technique can integrate a large amount of data and knowledge representing the same real-world object and obtain a consistent, accurate, and useful representation of that object. The data may be independent or redundant, and can be obtained by different sensors at the same time or at different times. A suitable combination of investigative methods can substantially increase the profit of information in comparison with that from a single sensor. Multi-sensor information fusion has been a key issue in sensor research since the 1970s, and it has been applied in many fields. For example, manufacturing and process control industries can generate a lot of data, which have real, actionable business value. The fusion of these data can greatly improve productivity through digitization. The goal of this special issue is to report innovative ideas and solutions for multi-sensor information fusion in the emerging applications era, focusing on development, adoption, and applications.

Information fusion technology has been in existence for several decades. At the beginning, this technology was mainly applied in military. The main reason is that at that time, the sensor was still a very expensive instrument. Military might be the only consumer who required superior performance without considering the cost.

In recent years, the application background of multi-sensor information fusion technology has undergone great changes. We have found that many civilian systems also have multi-sensor systems, such as unmanned vehicle and intelligent robot systems. Moreover, we have noticed that these systems have become a major part where the multi-sensor information fusion technology could be used, and they usually contain great research value. In these new application systems, multi-sensor information fusion technology also faces many new research issues. This is what the researchers are interested in this research field recently. We have discovered that there are many areas in which multi-sensor information fusion technology is worth investigating further.

Our special issue was consisted of 30 papers, including the latest research results of the multi-sensor information fusion technology.

In general, these research papers were mainly divided into two parts: one is theoretical research results, and the other is the application-oriented issues. The topic discussed in theoretical research involves in-depth research on methods and theories, and it proposes new methods. The related papers mainly included three aspects: (1) a new fusion method based on the Kalman filter, including the study of various nonlinear Kalman filters, such as CKF, UKF, etc.; (2) a method based on DS evidence theory; (3) new methods for images, including video tracking, expression recognition, etc. On the other hand, we are also delighted to see that there are many papers involving solutions for various applications, which also have extremely high reading and reference value.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Facial Expression Recognition with Fusion Features Extracted from Salient Facial Areas

Yanpeng Liu, Yibin Li, Xin Ma and Rui Song *

School of Control Science and Engineering, Shandong University, Jinan 250061, China;
liuyanpeng@sucro.org (Y.L.); liyb@sdu.edu.cn (Y.L.); maxin@sdu.edu.cn (X.M.)

* Correspondence: rsong@sdu.edu.cn

Academic Editors: Xue-Bo Jin; Shuli Sun; Hong Wei and Feng-Bao Yang

Received: 23 January 2017; Accepted: 24 March 2017; Published: 29 March 2017

Abstract: In the pattern recognition domain, deep architectures are currently widely used and they have achieved fine results. However, these deep architectures make particular demands, especially in terms of their requirement for big datasets and GPU. Aiming to gain better results without deep networks, we propose a simplified algorithm framework using fusion features extracted from the salient areas of faces. Furthermore, the proposed algorithm has achieved a better result than some deep architectures. For extracting more effective features, this paper firstly defines the salient areas on the faces. This paper normalizes the salient areas of the same location in the faces to the same size; therefore, it can extract more similar features from different subjects. LBP and HOG features are extracted from the salient areas, fusion features' dimensions are reduced by Principal Component Analysis (PCA) and we apply several classifiers to classify the six basic expressions at once. This paper proposes a salient areas definitude method which uses peak expressions frames compared with neutral faces. This paper also proposes and applies the idea of normalizing the salient areas to align the specific areas which express the different expressions. As a result, the salient areas found from different subjects are the same size. In addition, the gamma correction method is firstly applied on LBP features in our algorithm framework which improves our recognition rates significantly. By applying this algorithm framework, our research has gained state-of-the-art performances on CK+ database and JAFFE database.

Keywords: facial expression recognition; fusion features; salient facial areas; hand-crafted features; feature correction

1. Introduction

Facial expression plays an important role in our daily communication with other people. For the development of intelligent robots, especially indoor mobile robots, emotional interactions between robots and humans are the foundational functions of these intelligent robots. With automated facial expression recognition technology, these home service robots can talk to children and take care of older generations. Also, this technology can help doctors to monitor patients, which will save hospitals much time and money. In addition, facial expression technology can be applied in a car to identify whether the driver has fatigue, and this can save many lives. Facial expression recognition is worth researching because many situations need this technology. Many research works have been done in the literature; the universal expressions mentioned in papers are usually: anger, disgust, fear, happiness, sadness and surprise [1–3] while some researchers add neutral and contempt [4,5]. Different sensors are used to capture data of these expressions, and researchers recognize these basic expressions from two-dimensional (2D) and 3D spaces [6,7] faces. While different methods are applied to recognize the basic expressions in 2D and 3D spaces, landmarks localization processes are used in both 2D and 3D data. Vezzetti et al. [8,9] extracted many landmarks from multiexpression faces relying on facial

geometrical properties, which makes it easy to localize these parts on 3D faces. Many application scenarios, such as service robots, apply 2D images to detect and recognize faces, so our research focuses on recognizing expressions from 2D static images.

Different styles of data and various frameworks are applied to 2D space facial expressions recognition. Like other recognition research, facial expressions recognition uses data from videos, images sequences [10] and static images [3,11]. All of the movement processes of the expressions are applied in the research, which use videos and images. Research using static images only uses the peak frames because they contain sufficient information about the specific expressions, and that is also the reason why this paper chose to use the peak frames. There are two main kinds of algorithm frameworks applied in facial expressions recognition work. Algorithms that use mature descriptors such as Histogram of Oriented Gradient (HOG) [12] and Local Binary Patterns (LBP) [4] extract features from the images and then send the features to the classifiers. The performances of this kind of algorithm framework rely on the effectiveness of these descriptors. In order to fuse more effective descriptors, researchers extract different kinds of features and fuse them together [13]. Although the fusion features behave better than one kind of feature, these features' distinguishing features have not been fully used. Feature correction method is applied to the features in our paper and this significantly improves the recognition rate. Deep networks is another popular framework in the facial expression recognition domain. AU-inspired Deep Networks (AUDN) [2], Deep Belief Networks (DBN) [14] and the Convolutional Neural Network (CNN) [10,15] are used in facial expressions recognition work. Apart from the higher recognition rate, more computing resources and data are needed in these algorithms. For these reasons, the former framework is applied in our research.

Face alignment is applied to help researchers to extract more effective features from the static images [4,16]. Automated facial landmark detection is the first step to complete this work. After finding these landmarks on the faces, researchers can align the faces and extract features from these faces. Early days, for the limitation of face alignment technology, researchers use fewer landmarks to align the faces and separate the faces to several small patches for extracting features [13]. This can roughly align the faces while more landmarks can improve the alignment precision. There are many methods to detect landmarks from the faces. Tzimiropoulos et al. [17] proposed a Fast-SIC method for fitting AAMs to detect marks on the faces. Zhu et al. [18] use a model based on the mixture of trees with a shared pool which marks 68 landmarks on the face. This method is applied in our algorithm and 68 landmarks are used to align the salient areas. These landmarks mark the shape of the eyebrows, eyes, nose, mouth and the whole face, which can help researchers to cut the salient patches. Although alignment faces can help to extract more effective features from the faces, some areas on the faces do not align well during this process. In this paper, the idea of normalizing the salient areas is firstly proposed to improve the features' extracted effectiveness.

In order to reduce the features' dimensions and extract more effective features, different salient areas definite methods are proposed in the literature. Zhong et al. [3] explained the idea that discovering common patches across all the expressions is actually equivalent to learning the shared discriminative patches for all the expressions in their paper. They transferred the problem into a multi-task sparse learning (MTSL) problem and by using this method they obtained a good result. Happy et al. [13] applied these areas found in their paper and they also gained a decent result. Liu et al. [2] used Micro-Action-Pattern Representation in the AU-inspired deep networks (AUDN) and built four criterions to construct the receptive fields. This gained a better result and accomplished feature extraction at the same time. In order to define the salient areas more accurately, our research uses neutral faces to compare with the peak frames of these expressions. The Karl Pearson correlation coefficient [19] is applied to evaluate the correlation between the neutral faces and the faces that expressed different expressions. For finding the precise locations of the salient areas, the faces are separated to several small patches. After comparing the small patches to their neutral faces, the patches which have weaker correlation coefficient are found and these are the areas expressing the specific expression. By using this method, the salient areas of the six fundamental expressions are found and

after fusing these areas, the salient areas of the six basic expressions are found too. Landmarks of the faces are used to locate these salient areas and different sizes of salient areas are normalized in our research framework.

Different kinds of descriptors are applied in facial expressions recognition research. Regarding the scale of the features extracted areas, previously, hand-crafted features were extracted from the whole alignment face [4,16] but nowadays the salient areas are used in hand-crafted extraction [3,13]. Aiming to describe the different expressions more effectively, diverse features extracted methods are used in facial expression recognition. Typical hand-crafted features include Local Binary Patterns (LBP) [4], Histogram of Oriented Gradient (HOG) [12], Scale Invariant Feature Transform (SIFT) [20], and the fusion of these features [11]. According to the literature, the fusion features contain more information about these expressions and achieve better results. That is the reason why we chose to extract LBP and HOG from the faces. Although fusion features can improve recognition rate, it is hard to fuse these features well. Before different features fuse together, normalization methods are applied to the features. Although utilizing this normalization method can improve the recognition result, different kinds of features' identities cannot mix well. Aiming to use more information of the LBP features and normalize the LBP features, the gamma feature correction method is firstly applied on LBP features in our algorithm framework.

In this paper, a straightforward but effective algorithm framework has been proposed to recognize the six basic expressions from static images. The algorithm framework is shown in Figure 1. In order to define and obtain these salient areas from these faces, the faces and facial landmarks are detected in the first step. After doing that, these faces are separated into several patches and by comparing neutral faces to these expressions, the salient areas are defined. Until this step, the salient areas are separated from the faces according to these landmarks. For extracting more effective features from these salient areas, the idea of normalizing the salient areas is firstly proposed to overcome salient areas misalignment. After finishing that, LBP and HOG features are extracted from these salient areas. The gamma correction method is firstly applied on LBP features and then the classifier can use more information from these LBP features. The Z-score method is used to normalize the LBP and HOG features to fusing them. Before applying different classifiers to classify these six expressions, Principal Component Analysis (PCA) is utilized to reduce the dimensions. Finally, different classifiers are applied to evaluate the effect of our framework, and our framework has achieved a better grade than the deep networks [2,14].

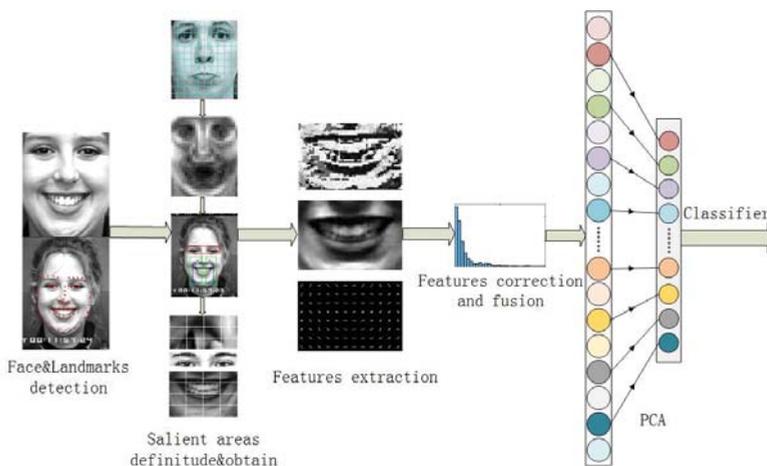


Figure 1. Framework of the proposed algorithm.

2. Methodology

In this section, the proposed algorithm will be explained in detail. This section will introduce the salient facial areas definitude principle and show the salient areas normalization and features fusion methods. LBP features correction methods will be introduced and applied in our algorithm. We will then introduce the following sections of this paper.

2.1. Faces Alignment and Salient Facial Areas Definitude

Automated face and facial landmark detection is the first step in our method. Facial landmark detection is an important base for facial expression classification. The method that is applied in the paper [18] is chosen to mark 68 landmarks on faces in our research. These landmarks mark the shape of the eyebrows, eyes, nose, mouth and the whole face, so these specific areas can be located by these landmarks. These 68 landmarks on the face and the normalized face have been shown in Figure 1. According to the average length and width of the faces and the proportion of the length and width, the faces in CK+ database are normalized to 240×200 .

As we all know, these six fundamental expressions have different salient areas. In this paper, an algorithm is proposed to find the salient areas in these expressions. For the purpose of extracting more effective features from the faces, people have applied different methods to calculate the salient areas in the faces. Zhong et al. [3] explained the idea that discovering the common patches across all the expressions is actually equivalent to learning the shared discriminative patches for all the expressions in the paper. Since multi-task sparse learning (MTSL) can learn common representations among multiple related tasks [21], they transferred the problem into an MTSL problem. They used this method and gained a good result. In order to learn expression specific features automatically, Liu et al. [2] proposed an AU-inspired deep network (AUDN). They used Micro-Action-Pattern Representation in the AUDN and built four criterions to construct the receptive fields. This gained a better result and accomplished features extraction at the same time.

These methods all found salient areas from the aligned faces and extracted features from these salient areas. As for our algorithm, the areas which are more salient to their own neutral faces are firstly found. In the last paragraph, the 68 landmarks have been found and the faces are normalized to 240×200 . The areas in the six basic expressions are compared to their neutral faces at the same location. If the areas during the expressions have not moved around, the areas must have more correlation with the areas on the neutral faces. Using this principle, compared to the other correlation coefficient methods in [19] the Karl Pearson correlation coefficient is applied to evaluate the correlation between the neutral faces and the faces expressed in different expressions. The Karl Pearson correlation coefficient is applied to evaluate the correlation between the matrixes. For finding the precise locations of the salient areas, the faces are separated to 750 (30×25) patches and every patch is 8×8 pixels. These 8×8 pixels patches are matrixes and by comparing the small patches from neutral faces and specific expressions, the salient areas can be precisely found. The Karl Pearson correlation coefficient formulate is shown next.

$$\gamma_{ij}^k = \frac{\sum_m (E_m - \bar{E})(N_m - \bar{N})}{\sqrt{(\sum_m (E_m - \bar{E})^2)(\sum_m (N_m - \bar{N})^2)}} \quad (1)$$

where γ_{ij}^k is the (i, j) th patch's correlation coefficient of specific expressions, so the scale of i is 1 to 30, j is 1 to 25. E_m is the pixel value of one subject from the specific expression and m ranges from 1 to 64 while N is the pixel value of the neutral face of that subject. \bar{E} is the mean of the small patch from specific expression, and \bar{N} is the mean of the neutral face.

$$R_{ij} = \sum_k \rho_k * \gamma_{ij}^k \quad (2)$$

In order to find the salient areas of all these six basic expressions, a formula is defined to evaluate the final correlation coefficient. The R_{ij} is the final correlation coefficient of the (i, j) th location on face.

By changing the ρ_k , the different proportion of the k th expression can be changed. Besides, the sum of the ρ_k must equal 1.

$$\sum_k^6 \rho_k = 1 \quad (3)$$

The results of the six expressions are shown in Figure 2. The areas found in this section will be applied in the next section. From Figure 2, we can find that different expressions have different salient areas. Equation (1) is used to evaluate the salient areas in the six fundamental expressions. In Equation (3), ρ_k expresses the proportion of the specific expression in the final result, the value of ρ_k can be changed according to the numbers of these different expressions because there are different numbers of images in these expressions.

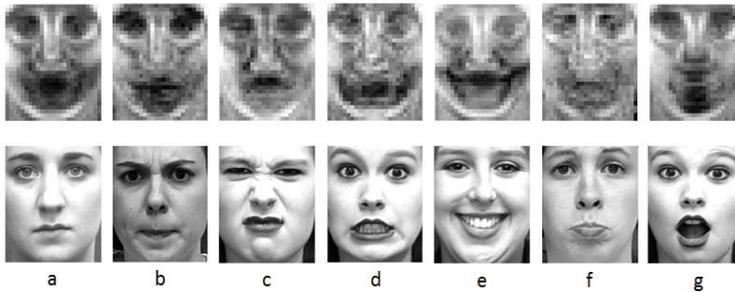


Figure 2. The salient areas of the six expressions. (a) Salient areas of all expressions and neutral; (b) Salient areas of anger; (c) Salient areas of disgust; (d) Salient areas of fear; (e) Salient areas of happy; (f) Salient areas of sad; (g) Salient areas of surprise.

2.2. Salient Areas Normalization and Features Extraction

In this section, the idea of normalizing the salient areas rather than the whole faces is proposed and applied. Furthermore, local binary patterns (LBP) features and the histogram of oriented gradient (HOG) features all are extracted from the salient areas. Compared to the method extracting features from the whole faces, features extracted from salient areas can reduce the dimensions, lower noise impacts and avoid overfitting.

2.2.1. Salient Areas Normalization

In the last section, these salient areas are determined. Our research has a similar result as the result in [3], but the performance differs in the eye areas. In papers [3,13], the researchers used the patches of the faces which come from alignment faces. Normalizing the whole faces is a good idea before more landmarks are marked from the faces. Then, more landmarks can be marked from the faces, which makes it easier to extract the salient areas from the faces. There are two main reasons for choosing to normalize the salient areas.

Firstly, aligning the faces may result in salient areas being misaligned. In order to demonstrate the alignment effects, the faces are aligned and then all the faces in one specific expression are added to gain the average face. The each pixel X_{ij} is the average of all images of the specific expression.

The salient mouth parts are separated from the faces and the average salient mouth areas are calculated for comparison. Figure 3 shows the result of the average faces, average salient areas and the mouth parts of the average faces. The mouth parts of the average faces are used to compare with the mouth parts using salient areas alignment. From the figure, it is clear that the mouth parts of the average faces have weaker contrast than the alignment mouth parts. This explains that aligning the faces leads to salient areas being misaligned. In contrast, by aligning the salient mouth areas, the mouths have a clear outline. Moreover, the alignment faces have different sized salient areas. Different

faces have different size, and the different sizes of the salient areas are extracted from these faces when we just use these landmarks to cut these salient areas. In the end, different dimensions of LBP and HOG are extracted from these salient areas. Using these features to classify the expressions can lead to worse recognition result. The reason why different LBP and HOG dimensions are extracted from these different sizes of salient areas can be explained by the principles of LBP and HOG which will be introduced in the next part. Aligning the whole faces may obtain different features from the same expression subjects because they have different sized areas to express the expression. This negatively influences the feature training in our algorithm framework. These are the reasons why the salient areas are normalized in our algorithm. Because our salient areas normalization method can overcome these shortcomings, our algorithm can gain a better result than only using the face alignment methods. In the experimental section, comparative experiments will be designed to compare the effects of the salient areas alignment method with the traditional faces alignment methods.

$$X_{ij} = \frac{1}{n} \sum x_{ij} \quad (4)$$

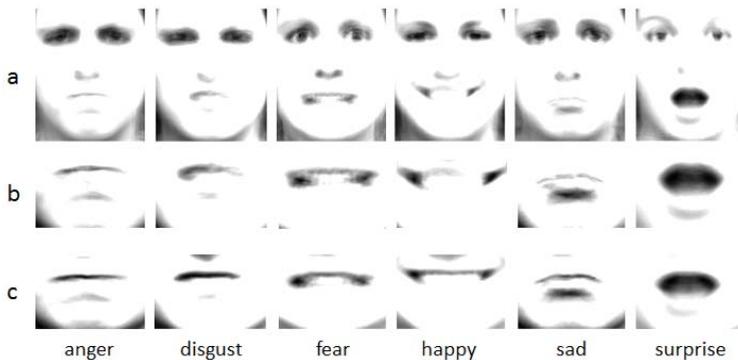


Figure 3. Average faces and the average salient areas. (a) Average faces of the six expressions; (b) Mouth areas of the average faces; (c) Average salient mouth areas.

2.2.2. Features Extraction

- Local Binary Patterns (LBP)

Texture information is an important descriptor for pattern analysis of images, and local binary patterns (LBP) were presented to gain texture information from the images. LBP was first described in 1994 [22,23] and from then on LBP has been found to be a powerful feature for texture representation. As for these facial expressions, actions of the muscles on the faces lead the faces to generate different textures. LBP features can describe the texture information of the images and this is the reason why LBP features are extracted from the salient areas. The calculation progress of the original LBP value is shown in Figure 4a. A useful extension to the original operator is the so-called uniform pattern [24], which can be used to reduce the length of the feature vector and implement a simple rotation invariant descriptor. In our research, a uniform pattern LBP descriptor is applied to gain features from the salient areas, and the salient areas are all separated to small patches. LBP features are gained from these salient areas respectively and these features are concatenated as the final LBP features. The length of the feature vector for a single cell can be reduced from 256 to 59 by using uniform patterns. This is very important, because there are many small patches in our algorithm. For example, the size of the mouth area is 40×60 and the small patches' size is 10×15 , so the mouth area is divided into $16(4 \times 4)$ patches. The uniform LBP features are extracted from each small patch and mapped to a

59-dimensional histogram. The salient areas all are separated into several small patches and the results are shown in Figure 5. The numbers are shown in Table 1. The dimension of the final LBP features is found to be 3068 by adding the numbers in Table 1.

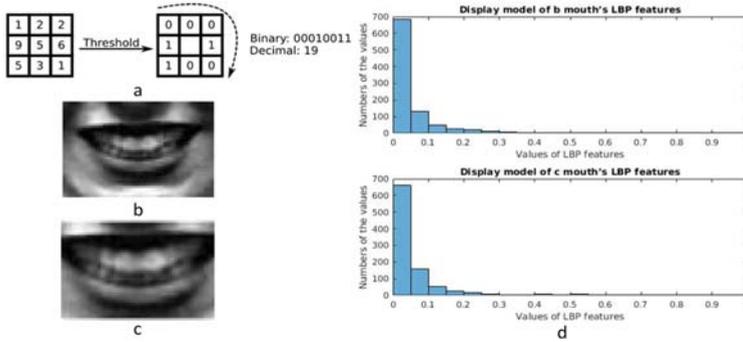


Figure 4. (a) Calculation progress of the original Local Binary Patterns (LBP) value; (b) Mouth area with pixels of 40×60 ; (c) Mouth area with the former pixels of 28×42 , the real pixels are enlarged from the former image; (d) Display models of (b,c) mouth.

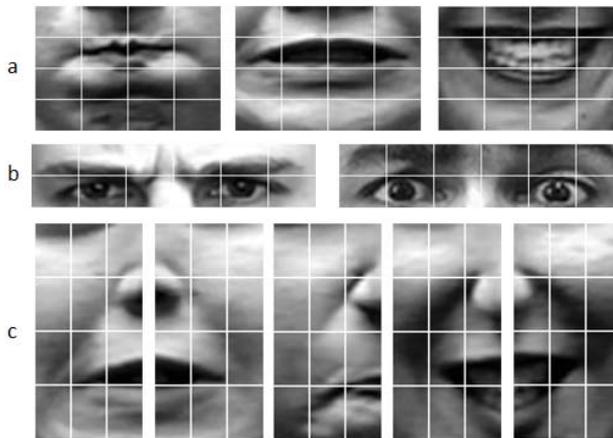


Figure 5. Small patches of salient areas. (a) Mouth areas, anger, fear, happy; (b) Forehead areas, anger, fear; (c) Cheek areas, left cheek fear, right cheek fear, left cheek anger, left cheek happy, right cheek happy.

Table 1. Patches numbers and LBP dimensions of salient areas.

Salient Areas	Forehead	Mouth	Left Cheek	Right Cheek
Pixels	20×90	40×60	60×30	60×30
Small patches number	12	16	12	12
LBP dimension	708	944	708	708
Total		3068		

Because different features will be extracted from different sizes of salient areas, these salient areas should be aligned. In order to demonstrate the difference, different sizes of mouth areas are cut from

one face and these areas are normalized to the same size. These mouth areas are shown in Figure 4. LBP features are extracted from these normalized faces and their distributions are shown in Figure 4. From the figure, we can know that the values in (d) and (e) result in different performance, and a conclusion can be drawn that different sizes of images have different LBP features. Besides, HOG features also are extracted from these salient areas, and they have a similar result as for LBP features.

- Histogram of Oriented Gradient (HOG)

Histogram of oriented gradients (HOG) is a feature descriptor which is used in computer vision and image processing [25]. The technique counts occurrences of gradient orientation in localized portions of an image. HOG descriptors were first described in 2005 [26], the writers used HOG for pedestrian detection in static images. During HOG features extraction, the image is divided into several blocks and the histograms of different edges are concatenated as shape descriptor. HOG is invariant to geometric and photometric transformations, except for object orientation. Because the images that in these databases have different light conditions and different expressions have different orientations in the eyes, nose, lips corners, as a powerful descriptor HOG is selected in our algorithm. In our paper, for extracting HOG features, every cell is 5×5 and $4(2 \times 2)$ cells make up a patch. The dimension of the mouth area is 60×40 and every cell has 9 features, so the dimension of the mouth area is 2772. The dimensions of the four salient areas are shown in Table 2. The HOG descriptors are shown in Figure 6 and the figure shows that the mouth areas of different expression have different HOG descriptors.

Table 2. Patches numbers and Histogram of Oriented Gradient (HOG) dimensions of salient areas.

Salient Areas	Forehead	Mouth	Left Cheek	Right Cheek
Pixels	20×90	40×60	60×30	60×30
Small patches number	51	77	55	55
HOG dimension	1836	2772	1980	1980
Total	8568			

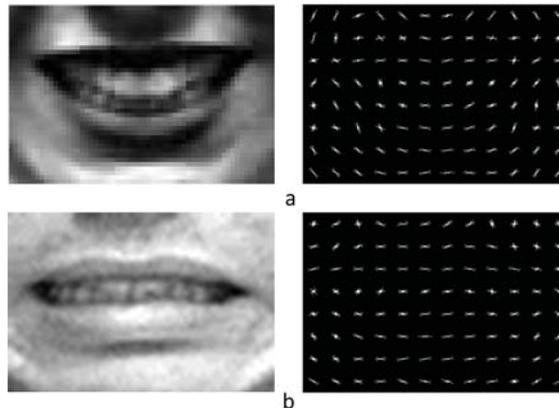


Figure 6. HOG descriptors of the mouths. (a) Happy mouth areas; (b) Fear mouth areas.

2.3. Features Correction and Features Fusion

2.3.1. LBP Correction

LBP feature is a very effective descriptor of the texture information of images. Many researchers [3,13] applied LBP to describe these different expressions. In our algorithm, LBP features are extracted and

processed before they are sent into the classifiers. For most papers, researchers extract features from images and normalize the data to 0–1, e.g., [27] normalizes the data to 0–1. For our algorithm framework, the image’s data are normalized to 0–1, and by using this method a better recognition rate can be gained. In order to normalize the LBP features of every subject, the method in Equation (5) is applied to normalize the LBP features to 0–1.

$$L_m = \frac{l_m}{\max(l_m)} \quad (5)$$

where m is the dimension number of every salient area and l_m is the value of the LBP feature.

Aiming to utilize the specific characteristic of every area, these four salient areas are normalized respectively. The distribution styles of these LBP features extracted from the four salient areas are displayed in Figure 7. The figure shows that the distribution model of LBP features is the power law distribution. In image processing, the gamma correction redistributes native camera tonal levels into ones which are more perceptually uniform, thereby making the most efficient use of a given bit depth. In our algorithm, the distribution of the LBP features is the power law distribution, and therefore more information concentrates on minority LBP features. According to our experiment, although the distributions of these subjects’ LBP features all follow the power law distribution, the specific values are slightly different. For example, LBP values from 0 to 0.5 have different numbers when different subjects’ images are processed. This can be changed by gamma correction method, and by doing this, more information can concentrate on more LBP features. For using more information from LBP features and making it easy to fuse LBP and HOG features, gamma correction is used to correct the LBP feature data.

$$\tilde{L}_m = L_m^{\frac{1}{\lambda}} \quad (6)$$

where λ is the correct gamma number. As is well known, most values of the gamma number come from experimental experience data. The parameter σ is proposed in our algorithm to help to find the proper λ . In Equation (7) we have defined the mathematical expression of σ . From Equation (7) we know that σ has a similar meaning to variance and the zero value is separated from these data because it has no meaning. The gamma correction method is proposed to use more information from the initial LBP features, and the value of variance shows the fluctuation of the data. For the power law distribution, fewer data contain more information. As for the LBP features when the gamma correction method is applied, more LBP features contain more information and therefore the fluctuation of LBP features becomes bigger. That is the reason why parameter σ is proposed and applied, and the relationship between σ and the gamma number λ proves the correctness of our method.

Every salient area is processed respectively, so four σ values will be gained. The relationships between these four salient areas’ σ and the gamma number λ are shown in Figure 8. According to the experimental data, all the four salient areas have the maximum σ value around λ ’s value 2. Therefore, all the four salient areas’ σ values are added to find the final sum and the related λ .

$$\sigma = \frac{1}{np} \sum_n \sum_p (L_{np} - U_n)^2 \quad (7)$$

$$U_n = \frac{1}{p} \sum_p L_{np} \quad (8)$$

where n is the number of the images’ number, and p is the number of nonzero data in LBP features of the salient areas. Therefore, p is smaller than the dimensions of the salient areas’ LBP features. L_{np} is the value of LBP feature and U_n is the mean value of LBP features from the specific subjects. In our algorithm, the relationship between the σ , gamma number λ and the recognition rate have been found, and the relationship is shown in Figure 9.

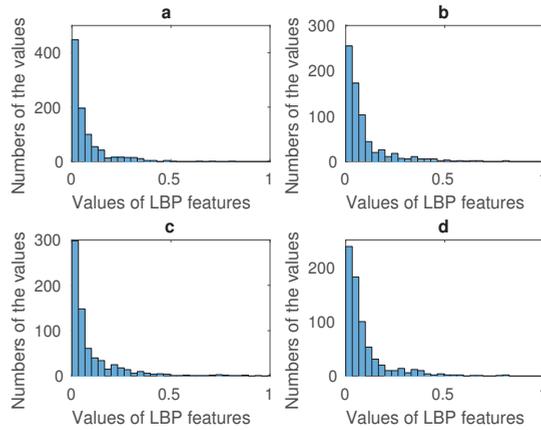


Figure 7. (a) Display model of the mouth’s LBP features; (b) Display model of the left cheek’s LBP features; (c) Display model of forehead’s LBP features; (d) Display model of the right cheek’s LBP features.

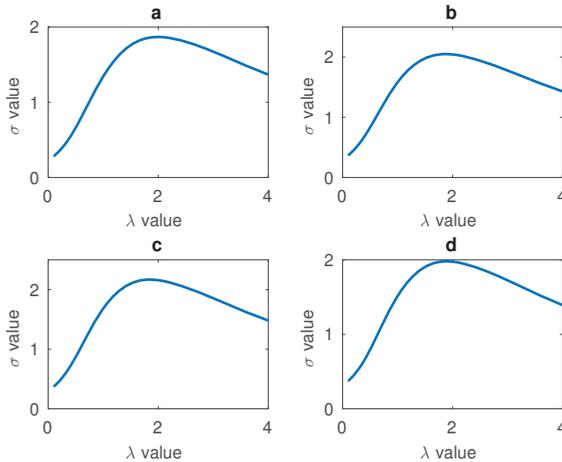


Figure 8. (a) Relationship between the mouth’s λ and σ ; (b) Relationship between the left cheek’s λ and σ ; (c) Relationship between the forehead’s λ and σ ; (d) Relationship between the right cheek’s λ and σ .

2.3.2. HOG Processing and Features Fusion

Different features describe different characters of the images; therefore, researchers have merged some features together to be able to take advantage of the superiority of all the features [13,27]. For our algorithm, the LBP and HOG descriptors are applied to utilize the texture and orientation information of these expressions. Proper fusion methods are very important factors for recognition work and unsuitable methods can make the recognition result worse. The recognition rate of the individual feature and fusion features will be shown in the experimental section. Zhang et al. [27] applied a structured regularization(SR) method which is employed to enforce and learn the modality specific sparsity and density of each modality, respectively. As for our algorithm, the single features are

firstly processed to their best performance and then they are normalized to the same scale. In the experimental section, different experiments are proposed to explain the results of single features and the fusion feature.

The gamma correction method is applied to ensure the LBP achieve their best performance. Different features must be processed to the same scale when these features are fused together. The Z-score method is used to process LBP and HOG features, and after applying the method the average is 0 while the variance is 1.

$$\sigma = \sum_j^J (f_j - \mu)^2 \quad (9)$$

$$\mu = \frac{\sum_j^J f_j}{J} \quad (10)$$

$$\hat{f}_j = K \frac{(x_j - \mu)}{\sigma + C} \quad (11)$$

where f_j is the data of LBP or HOG feature and \hat{f}_j is the feature data after processing. As for the LBP features, although the display model has changed, the data are changed to the same scale and a better result can be gained. Because \hat{f}_j is too small, number K is used to multiply \hat{f}_j . In our experiment, the K equal to 100.

2.4. Principal Component Analysis(PCA) and Classifiers

Principal Component Analysis (PCA) was invented in 1901 by Karl Pearson [28] as an analog of the principal axis theorem in mechanics, it was later independently developed by Harold Hotelling in the 1930s [29]. Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. PCA is an effective method to reduce the features dimension. There are many researchers using this method to reduce the features' dimension. In our algorithm, the fusion features' dimension is 11,636, which is really a very large number. In order to reduce the feature's dimension, PCA is applied. The relationship between the recognition date and the number of the dimension under softmax classifier is shown in Figure 10. According to the experiment, the most appropriate dimension is 80.

There are many kinds of classifiers applied in the facial expression recognition research and the researchers apply different classifiers to evaluate their algorithms. These classifiers include SVM with polynomial [4,13], linear [2,4,13] and RBF [4,13] kernel, and softmax [10,15] is also utilized in this work. In order to evaluate the effectiveness of our proposed algorithm and compare with the same work in other literature, many classifiers are applied in our experimental part. In our algorithm, different classifiers are applied to recognize these fusion features the dimensions of which are reduced by PCA.

3. Database Processing

3.1. CK+ Database

The CK+ database [5] is an extended database of the CK [30] database which contains both male and female subjects. There are 593 sequences from 123 subjects in the CK+ database, but only 327 sequences are assigned to 7 labels. These 7 labels are anger (45), contempt (18), disgust (59), fear (25), happy (69), sad (28), surprise (83). The sequences show the variation in the images from neutral to the peak of the expressions. The different expressions have different numbers in the sequences. In particular, the images extended in 2010 have different pixels and two types of pixels, which are 640×490 and 640×480 in the database. In order to compare with other methods [2,3,14,27], our experiments use these 309 sequences in the 327 sequences without contempt. Similar to the method used in [2,3], the first image (the neutral) and the last three peak frames are chosen for training and

testing. The ten-fold-validation method is applied in the experiments while the subjects are separated into 10 parts according to the ID of every subject. There are 106 subjects in the chosen database, so the subjects are distributed into 10 parts which have roughly equal image number and subject number.

3.2. JAFFE Database

The JAFFE database [31,32] consists of 213 images from 10 Japanese female subjects. Every subject has 3 or 4 examples of all the six basic expressions and also has a sample of neutral expression. In our experiment, 183 images are used to evaluate our algorithm.

4. Experiments

In this section the experimental setting and the details of our paper will be described. All comparison experiment ideas came from the second section of our paper and these experiments are applied to evaluate our methods and certify our algorithm's correctness. Our experiments are executed on CK+ and JAFFE database and the results are also compared with the recognition rates in the related literature.

4.1. Salient Areas Definitude Method Validation

The reasons why the salient areas rather than the whole faces are chosen in our algorithm have been introduced in Section 2. Experiments are designed to evaluate the performance of our salient areas' definitive method. In addition, the LBP features are extracted from all of the aligned faces and the aligned salient areas to gain the contrast recognition rates. These results are shown in Table 3. In addition, the salient areas are separated from the raw images rather than the alignment faces according to specific landmarks among the 68 landmarks on the face. In Table 3, the 10-fold cross validation method is used to evaluate the performance of our method, and in this case only the LBP features are used in our recognition experiment.

Table 3. Recognition rate on CK+ under different salient areas definitude methods.

Salient Areas Definitude Methods	Zhong 2012 [3] (MTSL)	Liu 2015 [2] (LBP)	Liu 2015 [2] (AUDN-GSL)	Proposed Method (with Gamma)	Proposed Method (without Gamma)
Classifier	SVM	SVM	SVM	SVM	SVM
Recognition rate	89.9	92.67	95.78	95.5	96.6

For the purpose of distinguishing that whether using the salient areas can be more effective than the whole face alignment methods or not, the mouth areas are normalized to 60×30 , the cheek areas are normalized to 30×60 while the eye areas are normalized to 20×90 . LBP features are extracted from the small patches whose sizes are 15×10 and then all these features are concatenated together. LBP features are used to evaluate the performance of our algorithm and compare the result with other methods. The results are shown in Table 3. Several comparison experiments are designed, SVM and classifier are applied to evaluate our algorithm. Comparing LBP features extracted from the alignment areas with the features extracted from salient areas on alignment faces and the whole alignment faces, better recognition rates can be gained by our algorithm by using the SVM classifier. Polynomial, Linear, and RBF kernel SVM are used in our experiment and the SVM classifier is designed by Chih-Chung Chang and Chih-Jen Lin [33]. The gamma correction method is used to process the LBP features in our experiment. Compared with the experiment designed by Zhong et al. [3] and Liu et al. [2], according to the results in Table 3, our algorithm has a more precise recognition rate.

4.2. Gamma Correction of LBP Features

In Section 2.3, a method was proposed to process the LBP features and the relationship between the σ and gamma number λ was found. In order to evaluate the effects of gamma correction and verify

the relationship between σ and gamma number λ some comparison experiments are designed in our paper. CK+ and JAFFE datasets are used in our experiments. In our experiments, the number of λ ranged from 0.1 to 3 and all the results are recorded. In order to show the relationship between σ , gamma correction number λ and the recognition rate clearly, some figures have been draw to display the trend of recognition rate and σ . In Figure 9 the σ is the sum of the four salient areas' σ in CK+ database. In the figure, while λ is equal to 1 there is no gamma correction, and from the figure one can see that the biggest recognition rate and the biggest σ value all result from the value of λ near to 1.8. The relationship between σ and λ of JAFFE database are shown in Figure 10. These two figures show that our LBP correction method has good universality power. In addition, because there are fewer images in the JAFFE database, we can see that the curves in Figure 10 are not smooth enough, but their overall trends also correspond with the relationship in Figure 10. The performances of these experiments are shown in Table 4, and different classifiers are used to evaluate the universality of the gamma correction method. Table 4 shows that gamma correction has significantly improved the recognition rate and this proves that our method of using σ to find the proper λ can be applied in facial expression recognition work. In Table 4, 10-fold cross validation is applied on the CK+ database and the leave-one-person-out validation method is used on the JAFFE database.

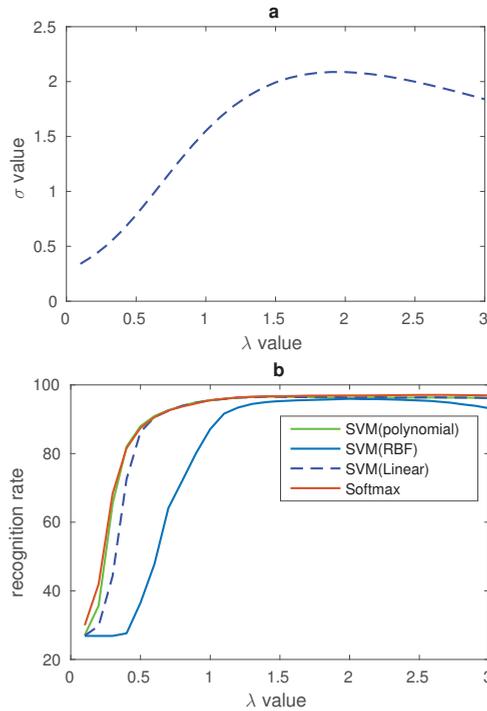


Figure 9. (a) Relationship between λ and σ on CK+ database; (b) Relationship between λ and recognition rates from different classifiers on CK+ database.

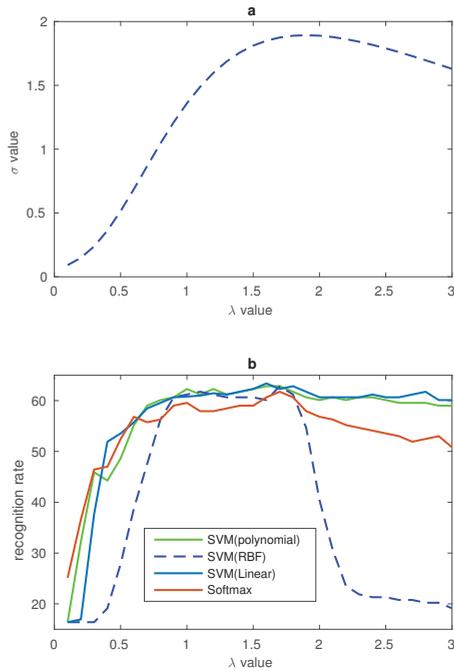


Figure 10. (a) Relationship between λ and σ on JAFFE database; (b) Relationship between λ and recognition rates from different classifiers on JAFFE database.

Table 4. Recognition rates on different classifiers with and without gamma correction.

	CK+		JAFFE	
	Gamma-LBP	LBP	Gamma-LBP	LBP
SVM(polynomial)	96.6	95.5	62.8	62.3
SVM(linear)	96.6	95.6	63.4	60.8
SVM(RBF)	96.0	87.1	62.8	61.2
Softmax	97.0	95.6	61.7	59.6

In order to compare our algorithm with other research, we apply the same classifier and validation method as in the literature. Compared with the literature in Table 5, our experiment on the CK+ database has better results and this shows that our salient areas definitude methods and LBP correction method have fine performance.

Table 5. Recognition rate on CK+ under LBP feature in different literature.

Methods	Zhong 2012 [3]	Shan 2009 [4]	Proposed Methods
Classifier	SVM	SVM	SVM
Validation Setting	10-Fold	10-Fold	10-Fold
Performance	89.9	95.1	96.6

4.3. Features Fusion, PCA and Recognition Rate Comparison

In our algorithm, LBP and HOG features are used to train the SVM and softmax classifiers and these features all are extracted from these salient areas. In order to gain a better result, LBP and HOG features are fused in our research. Using only the HOG feature, we obtain a 96.7 recognition rate and, using the fusion method, we obtain a better result, namely 98.3, which was reached on the CK+ database. In addition, a similar result has been obtained on the JAFFE database. Because using fusion features can lead to a better recognition result, fusion features are used in our algorithm.

The full dimension of fusion features is 11,636, which is a very large number. In addition, huge feature dimension can pull in some noise and lead to overfitting. As for the PCA method, if the number of the features' dimension is bigger than the images' number the principal component number is 1. In order to gain the most appropriate number, the number of the dimension is changed from 10 to 1000 and by using this method, the PCA dimension can be chosen according to recognition rate. Because using the softmax classifier can obtain better recognition rate than the other classifiers, we use softmax to show the effects of PCA. The relationship between PCA number and recognition rate is displayed in Figure 11. The most appropriate PCA dimension number is chosen according to the recognition rate and the dimension number of the features put into softmax is 80 on CK+ database and a similar curve is obtained by JAFFE.

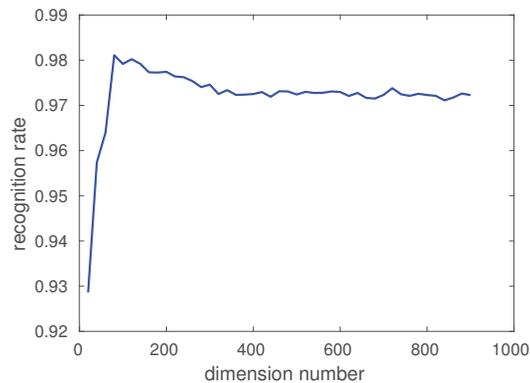


Figure 11. Relationship between Principal Component Analysis (PCA) dimension and recognition rate.

Until this step, the best recognition rate of 98.3 is gained under the 10-fold-cross validation method on the CK+ database. Therefore, to our knowledge, compared with the other methods in the literature, a state-of-the-art result has been obtained. Our result has been compared with other methods in the literature and the results are shown in Table 6. These four experiments all used deep networks while hand-crafted features are used in our algorithm. This explains that our algorithm has fine recognition ability by extracting features from the salient areas, correcting LBP features and fusing these features. In order to evaluate the adaptability of our algorithm, our algorithm also is applied on the JAFFE database. The results from other literature and our algorithm are shown in Table 7. The experiment shows that our algorithm has quite a good adaptability. Compared with the literature [13], our method can recognize about 5 more images than their method on average. In addition, compared with the literature [4], our features' dimension on CK+ and JAFFE is 11,636 which is much less than 16,640, and this illustrates that our algorithm needs less time and memory to train and predict.

Table 6. Recognition rate on CK+.

Literature	Liu 2014 [14]	Liu 2015 [2]	Jung 2015 [10]	Khorrani 2015 [15]	Proposed Algorithm
Method	BDBN	AUDN	DTAGN	Zero-bias CNN+AD	LBP+ HOG
Validation Setting	10-Fold	10-Fold	10-Fold	10-Fold	10-Fold
Accuracy	96.7	95.785	96.94	98.3	98.3

Table 7. Recognition rate on JAFFE.

Literature	Shan 2009 [4]	Happy 2015 [13]	Proposed Algorithm	Proposed Algorithm	Proposed Algorithm
Classifier	SVM(RBF)	SVM(Linear)	SVM(Linear)	SVM(Linear)	Softmax
Validation Setting	10-Fold	5-Fold	5-Fold	10-Fold	10-Fold
Accuracy	81.0	87.43	87.6	89.6	90.0

5. Discussion

More information about salient areas definitude methods is needed. Zhong et al. [3] designed a two-stage multi-task sparse learning algorithm to find the common and specific salient areas. LBP features rather than these pure data are used in this method, i.e., the LBP features are used to represent the information from the images. Liu et al. [2] built a convolutional layer and a max-pooling layer to learn the Micro-Action-Pattern representation which can depict local appearance variations caused by facial expressions. In addition, feature grouping was applied to find the salient areas. Compared with these two methods, our algorithm only uses the raw image data and there is no training procedure, but neutral faces are needed in our algorithm. For facial expressions, only partial areas in the face have changed, so the neutral faces can be used to calculate the correlation between neutral faces and specific expressions to localize the changed areas. Besides, the localiozation result can be more accurate when the changes are smaller. Furthermore, LBP features extracted from the small areas can also be used to compare and find the correlation. That is to say, by using these descriptors' property, our algorithm can be used to localize the changed areas.

Making the features extracted from different subjects in one class have a similar value is the main reason why gamma correction can improve our recognition rate. On the surface, gamma correction has changed the display of LBP features, but in fact it has changed the value of LBP features. Different subjects have different ways of presenting the same expression so their LBP features have some difference. According to our experiments, although their LBP features' values have some difference, their basic properties are similar. For instance, LBP features from the happy class have different values but these values are more different from those of the other classes. However, some subjects from different classes have similar LBP features and this is the reason why these algorithms cannot recognize the expressions. Our gamma correction method has shortened the distance in one class and this improves the recognition rate. The application of gamma correction on LBP features has had a positive effect on the recognition result and therefore some correction methods also can be applied on other features to shorten the distance in the same class to obtain a better recognition.

Although our algorithm has achieved a state-of-the-art recognition rate, there are some weakness in our method. Our algorithm selects these salient areas according to the landmarks on the faces, and if the landmarks are not accurate our recognition result will be influenced. Furthermore, if there is not enough image data, our gamma correction can not improve the recognition a lot. The performance of gamma correction on the JAFFE database shows these weaknesses. These are the main weaknesses of our algorithm.

6. Conclusions

The main contributions of this paper are summarized as follows: (1) A salient areas definitude method is proposed and the salient areas compared to neutral faces are found; (2) The idea of normalizing the salient areas to align the specific areas which express the different expressions is firstly proposed. This makes the salient areas of different subjects have the same size; (3) The gamma features correction method is firstly applied on the LBP features and this significantly improves the recognition result in our algorithm frameworks; (4) Fusion features are used in our framework, and by normalizing these features to the same scale, this significantly improves our recognition rate. By applying our algorithm framework, a state-of-the-art performance in the CK+ database under the 10-fold validation method using hand-crafted features has been achieved. In addition, a good result in the JAFFE database has also been obtained.

In the future, video data processing will be the focus of our research work and we will try to recognize facial expressions from real-time videos.

Acknowledgments: This work was supported in part by the National High Technology Research and Development Program 863, China (2015AA042307), Shandong Province Science and Technology Major Projects, China (2015ZDXX0801A02).

Author Contributions: Rui Song is the corresponding author who designed the algorithm and revised the paper. Yanpeng Liu conceived of, designed and performed the experiments, analyzed the data and wrote this paper. Yibin Li provided some comments and suggestions, and also revised the paper. Xin Ma provided some suggestions and comments for the performance improvement of the recognition algorithm.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Izard, C.E. *The Face of Emotion*; Appleton-Century-Crofts: New York, NY, USA, 1971.
2. Liu, M.; Li, S.; Shan, S.; Chen, X. Au-inspired deep networks for facial expression feature learning. *Neurocomputing* **2015**, *159*, 126–136.
3. Zhong, L.; Liu, Q.; Yang, P.; Liu, B.; Huang, J.; Metaxas, D.N. Learning active facial patches for expression analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 2562–2569.
4. Shan, C.; Gong, S.; McOwan, P.W. Facial expression recognition based on local binary patterns: A comprehensive study. *Image Vis. Comput.* **2009**, *27*, 803–816.
5. Lucey, P.; Cohn, J.F.; Kanade, T.; Saragih, J.; Ambadar, Z.; Matthews, I. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), San Francisco, CA, USA, 13–18 June 2010; pp. 94–101.
6. Wang, J.; Yin, L.; Wei, X.; Sun, Y. 3D facial expression recognition based on primitive surface feature distribution. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1399–1406.
7. Lei, Y.; Bennamoun, M.; Hayat, M.; Guo, Y. An efficient 3D face recognition approach using local geometrical signatures. *Pattern Recognit.* **2014**, *47*, 509–524.
8. Vezzetti, E.; Marcolin, F.; Fracastoro, G. 3D face recognition: An automatic strategy based on geometrical descriptors and landmarks. *Robot. Auton. Syst.* **2014**, *62*, 1768–1776.
9. Vezzetti, E.; Marcolin, F. 3D Landmarking in multiexpression face analysis: A preliminary study on eyebrows and mouth. *Aesthet. Plast. Surg.* **2014**, *38*, 796–811.
10. Jung, H.; Lee, S.; Park, S.; Lee, I.; Ahn, C.; Kim, J. Deep Temporal Appearance-Geometry Network for Facial Expression Recognition. *arXiv* **2015**, arXiv:1503.01532.
11. Zavaschi, T.H.; Britto, A.S.; Oliveira, L.E.; Koerich, A.L. Fusion of feature sets and classifiers for facial expression recognition. *Expert Syst. Appl.* **2013**, *40*, 646–655.
12. Hu, Y.; Zeng, Z.; Yin, L.; Wei, X.; Zhou, X.; Huang, T.S. Multi-view facial expression recognition. In Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, Amsterdam, The Netherlands, 17–19 September 2008; pp. 1–6.

13. Happy, S.; Routray, A. Robust facial expression classification using shape and appearance features. In Proceedings of the Eighth International Conference on Advances in Pattern Recognition (ICAPR), Kolkata, India, 4–7 January 2015; pp. 1–5.
14. Liu, P.; Han, S.; Meng, Z.; Tong, Y. Facial expression recognition via a boosted deep belief network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1805–1812.
15. Khorrami, P.; Paine, T.; Huang, T. Do Deep Neural Networks Learn Facial Action Units When Doing Expression Recognition? In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 7–13 December 2015; pp. 19–27.
16. Tian, Y. Evaluation of face resolution for expression analysis. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop, Washington, DC, USA, 27 June–2 July 2004; p. 82.
17. Tzimiropoulos, G.; Pantic, M. Optimization problems for fast aam fitting in-the-wild. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 593–600.
18. Zhu, X.; Ramanan, D. Face detection, pose estimation, and landmark localization in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 2879–2886.
19. Lee Rodgers, J.; Nicewander, W.A. Thirteen ways to look at the correlation coefficient. *Am. Stat.* **1988**, *42*, 59–66.
20. Tariq, U.; Lin, K.H.; Li, Z.; Zhou, X.; Wang, Z.; Le, V.; Huang, T.S.; Lv, X.; Han, T.X. Emotion recognition from an ensemble of features. In Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011), Santa Barbara, CA, USA, 21–25 March 2011; pp. 872–877.
21. Evgeniou, A.; Pontil, M. Multi-task feature learning. *Adv. Neural Inf. Process. Syst.* **2007**, *19*, 41.
22. Ojala, T.; Pietikainen, M.; Harwood, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Proceedings of the 12th IAPR International Conference Pattern Recognit. Conference A: Computer Vision & Image Processing, Jerusalem, Israel, 9–13 October 1994; Volume 1, pp. 582–585.
23. Ojala, T.; Pietikainen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* **1996**, *29*, 51–59.
24. Heikkila, M.; Pietikainen, M. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 657–662.
25. Wikipedia. Histogram of Oriented Gradients. Available online: https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients (accessed on 23 January 2017).
26. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
27. Zhang, W.; Zhang, Y.; Ma, L.; Guan, J.; Gong, S. Multimodal learning for facial expression recognition. *Pattern Recognit.* **2015**, *48*, 3191–3202.
28. Person, K. On Lines and Planes of Closest Fit to System of Points in Space. *Philosoph. Mag.* **1901**, *2*, 559–572.
29. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **1933**, *24*, 417.
30. Kanade, T.; Cohn, J.F.; Tian, Y. Comprehensive database for facial expression analysis. In Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France, 26–30 March 2000; pp. 46–53.
31. Lyons, M.; Akamatsu, S.; Kamachi, M.; Gyoba, J. Coding facial expressions with gabor wavelets. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 200–205.
32. Lyons, M.J.; Budynek, J.; Akamatsu, S. Automatic classification of single facial images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 1357–1362.
33. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27.



Article

Object Tracking Using Local Multiple Features and a Posterior Probability Measure

Wenhua Guo *, Zuren Feng and Xiaodong Ren

Systems Engineering Institute, State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, China; fzr9910@mail.xjtu.edu.cn (Z.F.); rxd@mail.xjtu.edu.cn (X.R.)

* Correspondence: wh.guo@stu.xjtu.edu.cn; Tel.: +86-29-8266-7771

Academic Editors: Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang

Received: 20 February 2017; Accepted: 28 March 2017; Published: 31 March 2017

Abstract: Object tracking has remained a challenging problem in recent years. Most of the trackers can not work well, especially when dealing with problems such as similarly colored backgrounds, object occlusions, low illumination, or sudden illumination changes in real scenes. A centroid iteration algorithm using multiple features and a posterior probability criterion is presented to solve these problems. The model representation of the object and the similarity measure are two key factors that greatly influence the performance of the tracker. Firstly, this paper propose using a local texture feature which is a generalization of the local binary pattern (LBP) descriptor, which we call the double center-symmetric local binary pattern (DCS-LBP). This feature shows great discrimination between similar regions and high robustness to noise. By analyzing DCS-LBP patterns, a simplified DCS-LBP is used to improve the object texture model called the SDCS-LBP. The SDCS-LBP is able to describe the primitive structural information of the local image such as edges and corners. Then, the SDCS-LBP and the color are combined to generate the multiple features as the target model. Secondly, a posterior probability measure is introduced to reduce the rate of matching mistakes. Three strategies of target model update are employed. Experimental results show that our proposed algorithm is effective in improving tracking performance in complicated real scenarios compared with some state-of-the-art methods.

Keywords: object tracking; multiple features; posterior probability measure; centroid iteration

1. Introduction

Among the numerous subjects in computer vision, object tracking is one of the most important fields. It has many applications such as human computer interaction, video analysis, and robot control systems.

Many object tracking algorithms have been brought up in the last decades. Welch [1] proposed a Kalman filter-based algorithm considering Gaussian and linear problems to track one's pose in interactive computer graphics. Later, a particle filter-based approach was introduced with respect to non-Gaussian and non-linear systems [2,3]. Other common trackers used include optical flow-based tracking [4], multiple hypothesis tracking [5,6], and kernel-based tracking [7,8]. Recently, João F. Henriques et al. [9] proposed a new kernel tracking algorithm called high-speed tracking with kernelized correlation filters that have been widely used. Unlike other kernel algorithms, the method has the exact same complexity as its linear counterpart.

Though these algorithms have been successful in many real scenes, they are still confronted with challenging problems, such as illumination changes, object occlusions, image noises, low illumination, fast motions and similarly colored backgrounds. One of the effective solutions is the mean-shift algorithm which can handle object partial occlusions and background clutters [10–12]. Mean-shift is a non-parametric pattern matching tracking algorithm. It uses the color histogram as the target model

and the Bhattacharyya coefficient as the similarity measure. The location of the target is obtained by an iterative procedure [10]. The performance of the algorithm is determined by the similarity measure and the target feature. Because of the background interference, the tracking result may easily get biased or be completely wrong. The location of the target obtained by the Bhattacharyya coefficient [7] or other similarity measures, such as normalized cross correlation, histogram intersection distance [13], and Kullback–Leibler divergence [14] may not be the ground truth. To improve the accuracy of object matching, a maximum posterior probability measure was proposed [15]. It takes use of the statistical feature of the searching region and can effectively reduce the influence of background and emphasize the importance of the target.

In some scenes with dramatic intensity or color changes, the effectiveness of the color decreases. Thus, it is desirable that some additional features should be used as a complement to color to improve the performance of the tracking system [16,17]. For example, Collins et al. [18] presented an online feature selection algorithm based on a basic mean-shift approach. The method can adaptively select the best features for tracking. They only used the RGB histogram in the algorithm, but it can be extended to other features. Wang et al. [19] proposed integrating color and shape-texture features for reliable tracking, and their method was also based on the mean-shift algorithm. Ning et al. [20] presented a mean-shift algorithm using the joint color-texture histogram, which proved to be more robust and insensitive than the color. Most of these methods used multiple features to describe the target model in order to reduce the mistakes of tracking systems. Unfortunately, color, shape-texture silhouettes or other traditional features can not track the target in some special scenes with variably scaled images or rotated images. In recent years, some new features have been proposed to solve these problems including Scale Invariant Feature Transform (SIFT) [21], Principal Components Analysis-Scale Invariant Feature Transform (PCA-SIFT) [22], Gradient Location and Orientation Histogram (GLOH) [23], Speed-up Robust Feature (SURF) [24], and Fast Retina Keypoint (FREAK) [25], just to name a few. Among them, a texture feature named the local binary pattern (LBP) [26] has been widely used in computer vision [27] due to its advantages of fast computation and rotation invariance. Recently, some improvements have been made based on the LBP such as the center-symmetric local binary pattern (CS-LBP) [28] and the local ternary pattern (LTP) [29].

This paper proposes a centroid iteration algorithm with multiple features based on a posterior probability measure [15] for object tracking. The main goal is to solve the difficulties in real scenes such as similarly colored backgrounds, object occlusions, low illumination color image and sudden illumination changes. The proposed algorithm consists of a target model construction step and a localization step. We improve the LBP descriptor to the DCS-LBP descriptor. For further improvement, a simplified version of the DCS-LBP is used, which we call the SDCS-LBP. It can describe important information of the image (the edge, the corner and so on). Then, this new texture feature and the color are combined to constitute the multiple features used in the target model, which we call the color and texture (CT) feature in this paper. After obtaining the target, three strategies for updating the target model are presented to reduce the tracking mistakes.

The rest of the paper is organized as follows: in Section 2, a local color texture feature based on the DCS-LBP along with its simplified form is introduced. In Section 3, the proposed tracking algorithm is illustrated in detail. Experimental results are shown in Section 4. Section 5 draws conclusions.

2. Multiple Features

Feature descriptors are very important in matching-based tracking algorithms, especially for applications in real scenes. In some simple scenes, color can work well because it distinguishes the targets from the background easily and contains a lot of useful information of the target. However, in complex scenes containing similarly colored backgrounds, object occlusions, low illumination color image and sudden illumination changes, the tracker only using the color feature may easily miss the target. One of the solutions is to integrate multiple features in the target model for reliable tracking.

2.1. Local Binary Patterns (LBPs)

The LBP is an illumination invariant texture feature. The operator uses the gray levels of the neighboring pixels to describe the central pixel. The texture model $LBP_{P,R}$ is expressed as follows [26]:

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c)2^i, \tag{1}$$

$$s(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

where P is the number of the neighbours and R is the radius of the central pixel. g_c denotes the gray value of the central pixel and g_i denotes that of the P neighbours with $i = 0, \dots, P - 1$, and $s(x)$ represents the sign function. Figure 1 gives an example of the LBP code when $P = 8$ and $R = 1$.

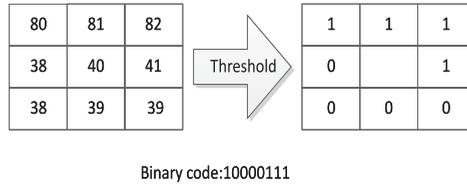


Figure 1. The original LBP code.

There are two extensions of the LBP [26]. The first one is to make the LBP as a rotation invariant feature as proposed by Ojala et al. [26]. It is defined as:

$$LBP_{P,R}^i = \min(ROR(LBP_{P,R}, i) | i = 0, 1, \dots, P - 1), \tag{2}$$

where $ROR(x, i)$ performs a circular bit-wise right shift on the P_bit number x by i times. Equation (2) selects the minimal number to simply the function. They explained that there were 36 rotation invariant LBP codes at $P = 8, R = 1$. The second one is the uniform LBP, which contains at most one 0–1 and one 1–0 transition when viewed as a circular bit string. The uniform LBP codes contain a lot of useful structural information. Ojala et al. [26] observed that although only 58 of the 256 8-bit patterns were uniform, nearly 90% of all observed image neighborhoods were uniform and many of the remaining ones contained noise. The following operator $LBP_{8,1}^{riu2}$ is a uniform and rotation invariant pattern with $Uvalue$ of at most 2:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{i=0}^{P-1} s(g_i - g_c)2^i, & U(LBP_{P,R} \leq 2), \\ P + 1, & \text{otherwise,} \end{cases} \tag{3}$$

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{i=1}^{P-1} |s(g_i - g_c) - s(g_{i-1} - g_c)|.$$

If we set $P = 8, R = 1$, the nine most frequent patterns with index from 0 to 8 are selected from the 36 different patterns, which are the rotation invariant patterns as shown in Figure 2.

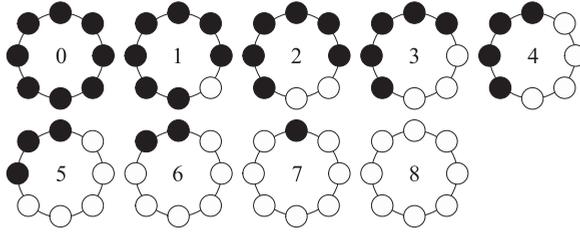


Figure 2. Nine uniform patterns of $LBPr_{8,1}^{2}$.

2.2. Center-Symmetric Local Binary Patterns (CS-LBPs) and Local Ternary Patterns (LTPs)

In Section 2.1, it can be seen that LBP codes have a long histogram, which require lots of calculations. Heikkilä et al. [28] designed a method by comparing the neighboring pixels in order to reduce computation. They calculated the center-symmetric pairs of the pixels as defined in the following function:

$$CS-LBP_{P,R} = \sum_{i=0}^{\frac{P}{2}-1} s(g_i - g_{i+\frac{P}{2}})2^i, \quad (4)$$

$$s(x) = \begin{cases} 1, & x \geq T, \\ 0, & \text{otherwise.} \end{cases}$$

This operator halves the calculations of LBP codes at the same neighbors. The LBP threshold depends on the central pixel, which makes the LBP sensitive to noise especially in flat regions of the image while the CS-LBP threshold is a constant value T that can be adjusted.

Tan et al. [29] extended the LBP to 3-valued codes, called the local ternary pattern (LTP). They set the codes around g_c in a zone of width $\pm T$ to one. The codes above it are set to 2 and the ones below it are set to 0. It is defined as:

$$LTP_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c)3^i, \quad (5)$$

$$s(x) = \begin{cases} 2, & x \geq T, \\ 1, & -T < x < T, \\ 0, & x \leq -T. \end{cases}$$

Here, T is the same threshold as the CS-LBP. Thus, the LTP is more insensitive to noise than the CS-LBP. However, it is no longer invariant to gray-level transformations.

2.3. Double Center-Symmetric Local Binary Patterns (DCS-LBPs)

In Section 2.2, it is analyzed that the CS-LBP is more efficient than the LBP in calculation, but they are both sensitive to noise. The LTP is insensitive to noise, but its computation is too complex. A simple way is to combine the LTP and the CS-LBP, which yields the CS-LTP. It is defined as:

$$CS-LTP_{P,R} = \sum_{i=0}^{\frac{P}{2}-1} s(g_i - g_{i+\frac{P}{2}})3^i, \quad (6)$$

$$s(x) = \begin{cases} 2, & x \geq T, \\ 1, & -T < x < T, \\ 0, & x \leq -T. \end{cases}$$

By definition, the CS-LTP retains the advantages of the CS-LBP and the LTP, but the ternary values are hard to calculate in the image.

Thus this motivates us to generate a DCS-LBP operator. The operator is divided into two parts: $DCS-LBP_{p,R}^{(upper)}$, in which the gray levels of the center-symmetric pixels above T are quantized to one while those below T are quantized to zero, and $DCS-LBP_{p,R}^{(lower)}$, in which the center-symmetric pixels on the other side below $-T$ are quantized to one while those below T are quantized to zero.

$$\begin{cases} DCS-LBP_{p,R}^{upper} = \sum_{i=0}^{\frac{p}{2}-1} s_1(g_i - g_{i+\frac{p}{2}})2^i, & s_1(x) = \begin{cases} 1, & x \geq T, \\ 0, & \text{otherwise,} \end{cases} \\ DCS-LBP_{p,R}^{lower} = \sum_{i=0}^{\frac{p}{2}-1} s_2(g_i - g_{i+\frac{p}{2}})2^i, & s_2(x) = \begin{cases} 1, & x \leq -T, \\ 0, & \text{otherwise.} \end{cases} \end{cases} \quad (7)$$

T is the threshold used to eliminate the influence of weak noise. The value of T determines the anti-noise capability of the operator. The upper-part and the lower-part of the DCS-LBP should be calculated separately and then be combined together for use. By definition, there are $2 \times 2^{\frac{p}{2}}$ different values, which are much less than the basic LBP (2^p) and the LTP (3^p), and are close to the CS-LBP ($2^{\frac{p}{2}}$) and the CS-LTP ($3^{\frac{p}{2}}$). When $P = 8, R = 1$, the DCS-LBP has 32 different values. Table 1 shows examples of all of these five local patterns. The first row are three local parts of an image including texture flat areas, texture flat areas with noise, and texture change areas. The threshold is set to be 5. It can be seen that the LBP and the CS-LBP can not exactly distinguish between texture flat and change areas. The other three patterns are distinguishable and are all insensitive to noise, among which the computational complexity of the DCS-LBP is lower than the other two.

It should be noted that there is a great amount of redundant information in the DCS-LBP, which might cause matching errors. Thus, further optimization is necessary. The DCS-LBP patterns also have the rotation invariant identity as shown in Figure 3. There are nine rotation invariant patterns. Similarly, both $DCS-LBP_{p,R}^{(upper)}$ and $DCS-LBP_{p,R}^{(lower)}$ have the same uniform patterns as the LBP. Pattern 5 to Pattern 8, which cannot describe the primitive structural information corresponding of the local image, are not uniform patterns. Pattern 0 to Pattern 4 each has its identity. Pattern 0 and Pattern 1 represent noise points, dark points and smooth regions. Pattern 2 represents the terminal. Pattern 3 represents angular points. Pattern 4 represents boundary. Thus, we improve the DCS-LBP to its simplified version (called SDCS-LBP), which retains only the patterns with index from 0 to 4.

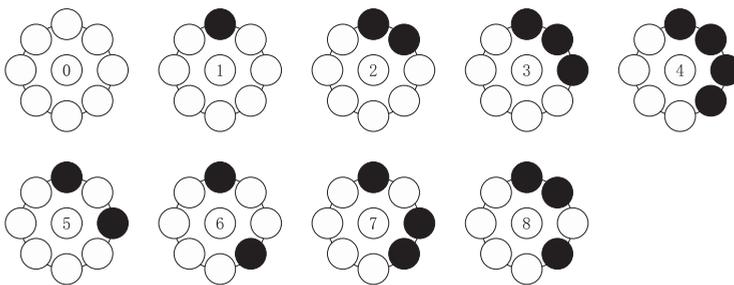


Figure 3. The nine rotation invariant patterns of the DCS-LBP.

Table 1. Examples of five coding rules ($T = 5$).

Image local region			
	Texture flat areas	Texture flat areas with noise	Texture change areas
LBP pattern			
	$[1111111]_2$	$[1000011]_2$	$[1000011]_2$
CS-LBP pattern			
	$[0000]_2$	$[0000]_2$	$[0000]_2$
LTP pattern			
	$[1111111]_3$	$[1111111]_3$	$[2111122]_3$
CS-LTP pattern			
	$[1111]_3$	$[1111]_3$	$[0001]_3$
DCS-LTP pattern			
	$[0000]_2[0000]_2$	$[0000]_2[0000]_2$	$[1000]_2[0011]_2$

2.4. Local Color Texture Feature (CT Feature)

Feature representation of the target model is very important for mean-shift based tracking algorithms. The original mean-shift algorithm selects the RGB color space ($16 \times 16 \times 16 = 4096$) as the features. However, in real scenes which contain similarly colored background, object occlusion, low illumination color image and sudden illumination changes, the original mean-shift algorithm can not track the target continuously. Inspired by [16], we consider designing a new feature combining the color and the texture.

This paper chooses to use the HSV color space, which contains Hue, Saturation and Value. The Value, which is measured with some white points, is often used for description of surface colors and remains roughly constant even with brightness and color changes under different illuminations. Hence, we replace the Value with the SDCS-LBP in the HSV space as the target model. The new feature which combines the color and the texture is called the CT feature in this paper. The CT feature can be considered as a special texture feature (terminal, angular point, boundary and some special points) with a certain color. The HSV color space is reduced to the size of 8×8 after excluding the part of the Value. Thus, the dimension of the CT feature is 640 ($8 \times 8 \times 5 \times 2 = 640$). Figure 4 shows three target models. For the CT feature, Figure 4b,c is the same and are different from Figure 4a, which can not be distinguished using the color alone. The CT feature has the rotation invariant identity and can distinguish between different texture patterns.

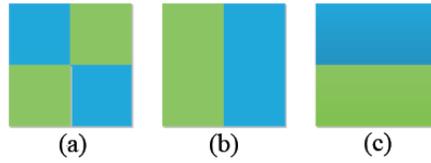


Figure 4. A particular target model.

The calculation process of the CT feature is as follows. Firstly, let P_i be the set of pixels of the target. Calculate $DCS-LBP_{P,R}^{upper}$, $DCS-LBP_{P,R}^{lower}$ and the HSV color space of each point in P_i in turn. If the value of $DCS-LBP_{P,R}^{upper}$ or $DCS-LBP_{P,R}^{lower}$ does not belong to the SDCS-LBP, the point will be seen as a meaningless point, which should be eliminated. Secondly, calculate $CT_{P_i}^{upper}$ and $CT_{P_i}^{lower}$ by multiplying the SDCS-LBP, the Hue and the Saturation. Third, after all the points of the target have been calculated, $his^{upper}(H, S, T)$ and $his^{lower}(H, S, T)$ of the target are worked out by putting the CT feature into the histograms. The histogram of the target model ($his(CT)$) is obtained by combining $his^{upper}(H, S, T)$ and $his^{lower}(H, S, T)$. Figure 5 shows the representation of a target model by the proposed method. Figure 5a is the first frame of a sequence. The target is showed in Figure 5b. The histogram of the CT feature is showed in Figure 5c.

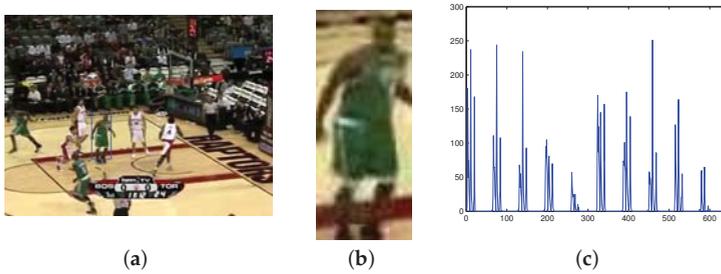


Figure 5. The representation model of the target by the proposed algorithm. (a) 1st frame; (b) target model region; (c) the histogram of CT feature.

3. Tracking Algorithm Using the CT Feature

Recently, many similarity measures are used in object tracking algorithms, such as the Euclidean distance, the Bhattacharyya coefficient, the histogram intersection distance, and so on. However, there is still lots of mismatching or misidentification in the tracking process. One of the reasons is that the target model contains some background pixels [15]. This paper proposes using the similarity measure based on maximum posterior probability to solve the problem.

3.1. Maximum Posterior Probability Measure

By introducing the candidate area, the maximum posterior probability measure (PPM) is able to decrease the influence of background and increase the importance of the target model in the tracking process. The PPM is a function to evaluate the similarity of the candidate and the target defined as:

$$\rho(p, q) = \frac{1}{m} \sum_{u=1}^{m_u} \frac{p_u q_u}{s_u}, \quad (8)$$

where p_u and q_u are, respectively, the histogram features of the target candidate region and the target model; s_u is the feature of the search region of the target candidate; m is the pixel number of the target model with $u = 1, \dots, m_u$; and m_u is the dimension of feature.

Now, we define a vector ω , which is computed according to Equation (9). $u(j)$ is the feature of the j th pixel; ω_j is the PPM of the j th pixel of the search region; A_i is the set of pixel of the i th target candidate region in the search region. Thus, the original PPM can be converted into a simple one as [15]:

$$\rho(p^i, q) = \frac{1}{m} \sum_{j \in A_i} \omega_j, \quad (9)$$

$$\omega_j = \begin{cases} \frac{q_{u(j)}}{s_{u(j)}}, & s_{u(j)} > 0, \\ 0, & s_{u(j)} = 0. \end{cases}$$

From the function, it can be found that the PPM and ω_j have a linear relationship. Therefore, we compute the incremental part to obtain the PPM of neighborhood, which makes the recursive algorithm a suitable one.

According to Equation (9), the PPM value of each pixel will be calculated, respectively. Thus, the matching process is simplified to find a target candidate region with the biggest sum of PPM value. The similarity measure of the target candidate and the target model is:

$$\rho_{y_i} = \sum_{x_i \in A_{y_i}} g(x_i), \quad (10)$$

$$g(x_i) = \frac{q_u(x_i)}{s_u(x_i)},$$

where $\{x_i\}_{i=1, \dots, m}$ is the set of pixel's position with the present frame centered at y_i ; $g(x_i)$ is the PPM value at x_i ; and A_{y_i} is the target candidate centered at y_i . Supposing the PPM value of each pixel as density and the similarity of the target candidate region as mass, the center of mass y_{i+1} is the target:

$$y_{i+1} = \frac{\sum_{x_i \in A_{y_i}} x_i g(x_i)}{\sum_{x_i \in A_{y_i}} g(x_i)}. \quad (11)$$

Figure 6 shows the PPM of the target model. The target bounded by the blue box and the target candidate region bounded by the green box in Figure 6a are resized. The target model and the target candidate region are showed in Figure 6b. The PPM of the target model, which holds monotonic and distinct peak shapes, is showed in Figure 6c.

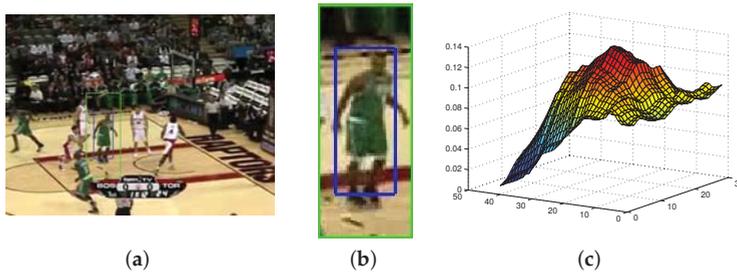


Figure 6. The maximum posterior probability of the target model. (a) 1st frame; (b) target candidate region; (c) the PPM of target model.

3.2. Scale Adaptation and Target Model Update

During the tracking process, the target always changes in shape, size, or color. Thus, the target model must be updated. The update must abide by certain rules to prevent the tracking drift. Three strategies are proposed for the target model update.

1. Introduce an adaptive process to fit the target region to a variable target scale for the purpose of precise target tracking.
2. Compute the similarity measure of the scale adapted target. If it is greater than a parameter, update the target model.
3. Introduce a parameter into the tracking algorithm to update part of the target model.

Strategy 1 introduces a scale adaptation function given by [15]:

$$\omega(k+1) = \begin{cases} \omega(k+2) + 2, & \text{if } \bar{\phi}_{-1} > 0.8 \text{ and } \bar{\phi}_0 > 0.75 \text{ and } \bar{\phi}_1 > 0.6, \\ \omega(k-2) - 2, & \text{if } \bar{\phi}_0 < 0.6 \text{ and } \bar{\phi}_1 < 0.3, \\ \omega(k), & \text{otherwise,} \end{cases} \quad (12)$$

where $\omega(k)$ is the size of the target region at frame k . $\bar{\phi}_i (i = -a, \dots, 0, \dots, a)$ is the average of the PPM of each pixel. Furthermore, $i < 0$ means the i th outer layer. $i = 0$ represents the target region border. a is the comparison step of scale adaptation and is set to 1 without losing the generality. In Equation (12), the expanding condition means the pixels around the border are likely to be a part of the target. The contracting condition means the target region should be reduced consequently. The function is an empirical one. The parameters should be trained by a great number of experiments.

Strategy 2 shows that the frame will not be updated until the similarity measure is greater than a certain parameter. In real scenes, some sudden changes may cause the tracking drift, so the update can not work every frame. p is the current frame model, while q is the target model. $\phi(p, q)$ is the similarity of the PPM for the current frame and the target model. If Equation (13) is satisfied, we considered p as the reliable CT feature model, and update the target model with p :

$$\bar{\phi}(p, q) \geq \delta. \quad (13)$$

Strategy 3 introduces a parameter into the algorithm to prevent the target model from being updated completely. Because of the limitations of the description to the target model, p can not take the place of q . The γ parameter is used to partially update the target model:

$$q' = \gamma p + (1 - \gamma)q, \quad (14)$$

where γ is the update factor; and q' is the updated CT feature model. In our experiment, γ is set to be a small value to adapt the changes of the target slowly.

3.3. Tracking Algorithm

Initialization: select the target object and compute the histogram $his(C, T)$ of the target model as q_u . The center of the target y_i is the initial position of the tracking object. Let $\{x_i\}_{i=1, \dots, m}$ be the set of pixel's position with the present frame centered at y_i .

1. Set y_i as the initial position. Calculate $his(C, T)$ of the search region as S_u .
2. Calculate the PPM values $g(x_i)$ of each pixel of the region by Equation (10).
3. Initialize the number of iterations as $k = 0$.
4. Calculate the target location by Equation (11). $k = k + 1$.
5. Repeat Step 4 until $\|y_{i+1} - y_i\| < \varepsilon$ or $k \geq N$.
6. Adjust the scale of the target region by Equation (12)
7. Decide whether to update the target by Equation (13). If satisfied, update the target model by Equation (14).
8. Read the next frame of the sequence and turn to Step 1.

If the distance between two iterations is less than ε or the number of iterations exceeds N , the algorithm is considered converged.

4. Experiments

The environments are set in some real scenes with similarly colored backgrounds, object occlusions, low illumination color image, and sudden illumination changes [12]. Eight public test sequences are used in experiments which are from the Visual Object Tracking challenge (<http://votchallenge.net/index.html>) and the Visual Tracker Benchmark [30] (<http://www.visual-tracking.net>) (see Figure 7). As the visual tracking benchmark, the test sequences are tagged with the following four attributes: low illumination color image (LI), sudden illumination changes (IC), object occlusion (OC), similarly colored background (SCB) (see Table 2). We designed a tracking system based on Matlab R2014a (8.3.0.532). All the trackers run on a standard PC (Intel (R) Core (TM) i5 2.6 GHz CPU with 8 GB RAM).

We compared our algorithm with some state-of-the-art methods including classical mean-shift tracking (KBT) [10], PPM-based color tracking algorithm (PPM) [15], a mean-shift algorithm using the joint color-texture histogram (LBPT) [20] and high-speed tracking with kernelized correlation filters (KCF) [9]. In addition, extra experiments are designed to test the function of the two major parts of the proposed method—the CT feature and the PPM separately. One of the experiments that we use is the CT feature with the Euclidean distance (CT&ED) instead of the PPM as the similarity measure. The other one that we use is the LBP feature with the PPM (LBP&PPM) instead of the CT feature. Both of the two trackers are tested in the experimental framework. All the methods aim at tracking one object in our experiments. The target will be tracked continuously at the rest of the frames.

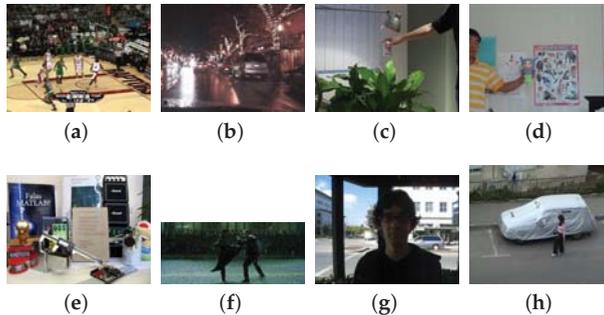


Figure 7. Eight test sequences used in current evaluation. (a) basketball; (b) car; (c) coke; (d) doll; (e) lemming; (f) matrix; (g) trellis; (h) woman.

Table 2. Eight sequences in the experiment.

Sequences	Size	Frame	fps	Object Number	Attributes
car	320 × 240	368	30	1	IC SCB LI
basketball	576 × 432	725	30	>8	IC OC SCB
coke	640 × 480	291	30	1	IC OC SCB
doll	400 × 300	3872	30	1	IC OC
lemming	640 × 480	1336	30	1	IC OC
matrix	800 × 336	100	30	2	IC OC SCB LI
Trellis	320 × 240	569	30	1	IC SCB LI
woman	352 × 288	597	30	1	IC OC

4.1. Parameter Setting

The size of the search region of our methods is set to 2.5 times the target size. In addition, there are five parameters in our tracking algorithm. We set $\delta = 0.85$ and $\gamma = 0.1$ for the target model update in Section 3.2. δ is the control parameter used to determine whether update the model or not. N and ϵ are the iteration parameters for the tracking algorithm in Section 3.3. $N = 20$ is the maximum number

of the iteration, and $\epsilon = 0.5$ is the minimum threshold of the iteration. The threshold parameter T is important in our algorithm. In order to test the sensitivity of the parameter, the central location error (CLE) is used to describe the tracking result. The CLE is defined as the Euclidean distance between the center of the box predicted by the tracker and that of the box of the ground truth. We set $T = 1, 3, 5, 7, 9$ for the calculation of the DCS-LBP. The results of eight test sequences are showed in Table 3. It can be seen that our algorithm performs well on all the tests when T is a small value between 1 to 5. In addition, it only missed the target in the basketball test sequence when T gets larger. Therefore, we set $T = 1$ in the experiments.

Table 3. The parameter setting (CLE).

SEQUENCE	T = 1	T = 3	T = 5	T = 7	T = 9
basketball	7	21	20	278	255
car	25	27	27	27	25
coke	19	18	17	14	16
doll	26	27	23	25	26
lemming	21	20	20	21	22
matrix	23	24	24	24	24
Trellis	13	13	12	12	12
woman	10	7	9	11	8
Average CLE	18	20	19	52	49

4.2. Qualitative Comparison

Some key frames of each sequence are given in Figure 8. The results of different trackers are shown by the bounding boxes in different colors.

- (1) In the basketball sequence, the tracked player moves fast. The environment changes many times. CT&ED lose the target at frame 80. KBT, PPM, and LBP&PPM fail at frame 473, when the player goes through his partner. KCF, LBPT and our tracker can successfully locate the object.
- (2) In the car sequence, the target is a car, but the road environment is dark. There are bright lights in the background. All of the trackers can merely track the car in the first 200 frames. However, at frame 260, the car turns right, and only KCF can track the car accurately.
- (3) In the coke sequence, the target is a coke and the light changes three times. The coke moves fast and is blocked by plants sometimes. When the coke is blocked by the plants the first time, LBTP misses the target. At frame 221, the occlusion and the illumination happen at the same time, and KBT and PPM obtain the wrong place. During the tracking, both KCF and our method perform better than the others.
- (4) The doll sequence has 3872 frames, which is a very long sequence. The target is a doll. It is blocked by the hand, and the scale of it changes sometimes. Because of the similar color with the background, LBP&PPM, LBPT, and CT&ED fail at frame 2378. KCF gives the best result followed by PPM and our tracker.
- (5) The lemming sequence is a challenging situation with fast motion, significant deformation and long-term occlusion. KCF missed the target at frame 380 because the target moves fast with the similar background. Our method is more effective than the others during the tracking.
- (6) In the matrix sequence, the target is the head. The sequence contains low illumination color image, sudden illumination changes, object occlusion, and similarly colored background. Our tracker gives the best result. At frame 30, all of the methods except ours lose the target. Our tracker misses the target at frame 90, when the target has dramatic changes in shape.
- (7) In the trellis sequence, the target is a boy's face in an outdoor environment. The situation has severe illumination and poses changes. All trackers except KCF and our tracker show some drifting effects at frame 270. The CT&ED loses the target at frame 410. Only KCF and our tracker show a good performance along the whole sequence.

- (8) In the woman sequence, the track is a walking woman in the street. The difficulty lies in the fact that the woman is greatly occluded by the parked cars. All the tracks fail at frame 124 except KCF and our tracker because of the occlusion and the small size of the target.

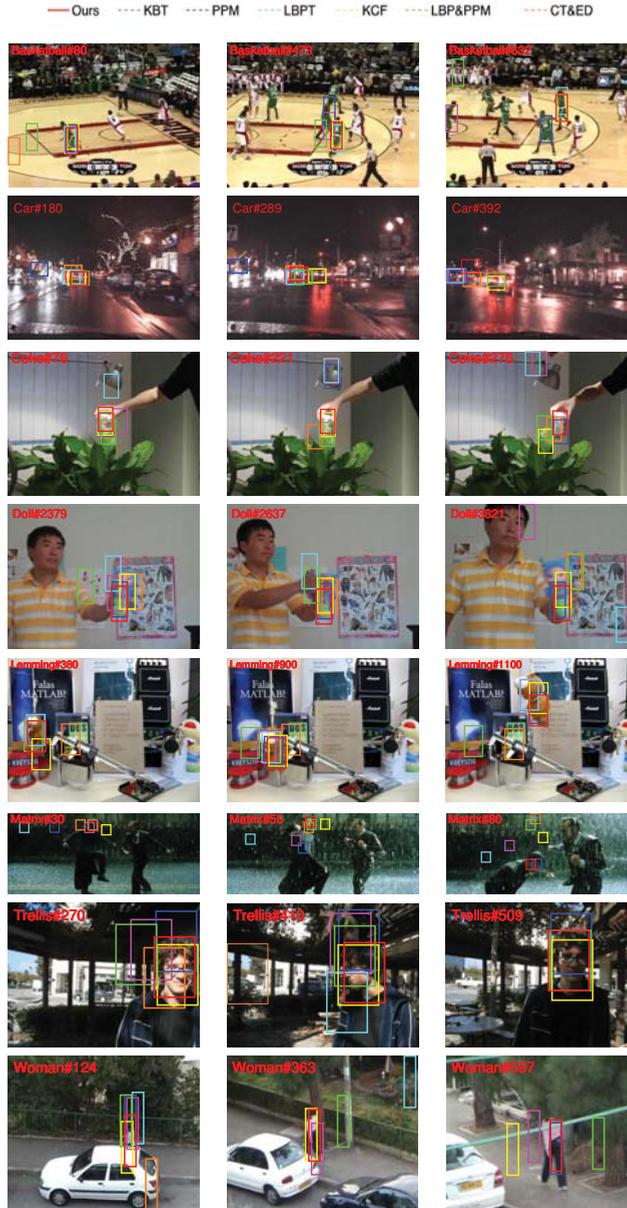


Figure 8. Experiment results of our proposed algorithm, KBT [10], PPM [15], LBPT [20], KCF [9], LBP&PPM and CT&ED on eight challenging sequences (from top to bottom are Basketball, Car, Coke, Doll, Lemming, Matrix, Trellis, Woman, respectively).

4.3. Quantitative Comparison

For performance evaluation and comparison, two metrics are considered: the CLE and the success rate (SR), which have been widely used in object tracking [12,31]. A target is considered as successfully tracked if the overlap region between the predicted bounding box and the ground truth exceeds 50% in a frame [32]. The SR is defined as

$$SR = \frac{area(M_t \cap M_g)}{area(M_t \cup M_g)}, \quad (15)$$

where M_t is the bounding box predicted by the tracker. M_g is the ground truth bounding box. The function $area(\bullet)$ means to calculate the area of a region. The CLE has been described in Section 4.1. The results of different methods on eight test sequences are showed in Tables 4 and 5. It can be seen from Tables 4 and 5 that our algorithm achieves an SR of 94% and a CLE of 18 which are better than the other algorithms. We also report the central-pixel errors frame-by-frame for each video sequence in Figure 9.

Now, we discuss the influence of the two major parts in our method: the CT feature and the PPM, separately. First, to test the influence of the similarity measure, we compare the trackers using the CT feature and different measures: the Euclidean distance (CT&ED) and the PPM (which is the proposed method—CT&PPM). It can be seen from Tables 4 and 5 that the PPM achieves an SR of 94% and a CLE of 18, which are better than those achieved by the Euclidean distance (40% and 122%). Second, to test the influence of the feature, we compare the trackers using the PPM and different features: the color feature (PPM), the LBP (LBP PPM) and the CT feature (which is the proposed method—CT&PPM). It can be seen from Tables 4 and 5 that the CT feature outperforms the others with the highest SR and a lowest CLE. The results demonstrate the effectiveness of both the CT feature and the PPM in improving the tracking accuracy.

Table 4. Success rates (%) of the proposed method compared with the other trackers.

SEQUENCE	KBT [10]	PPM [15]	LBPT [20]	KCF [9]	Proposed	LBP&PPM	CT&ED
basketball	65	68	100	100	100	56	3
car	65	20	63	100	71	76	51
coke	18	37	7	94	94	48	89
doll	88	100	79	100	97	57	56
lemming	99	99	83	68	100	38	24
matrix	41	15	7	31	91	57	49
Trellis	67	90	27	100	100	87	27
woman	93	53	19	94	95	42	18
Average success rate	67	60	48	86	94	58	40

Table 5. Center location errors of the proposed method compared with the other trackers (pixels).

SEQUENCE	KBT [10]	PPM [15]	LBPT [20]	KCF [9]	Proposed	LBP&PPM	CT&ED
basketball	113	68	11	8	7	123	288
car	29	77	31	6	25	16	36
coke	119	99	153	19	19	64	31
doll	25	12	42	8	26	51	67
lemming	13	12	61	78	20	149	132
matrix	75	14	249	76	23	61	85
Trellis	54	26	123	8	13	30	142
woman	22	85	145	10	10	46	196
Center location error	56	49	102	27	18	66	122

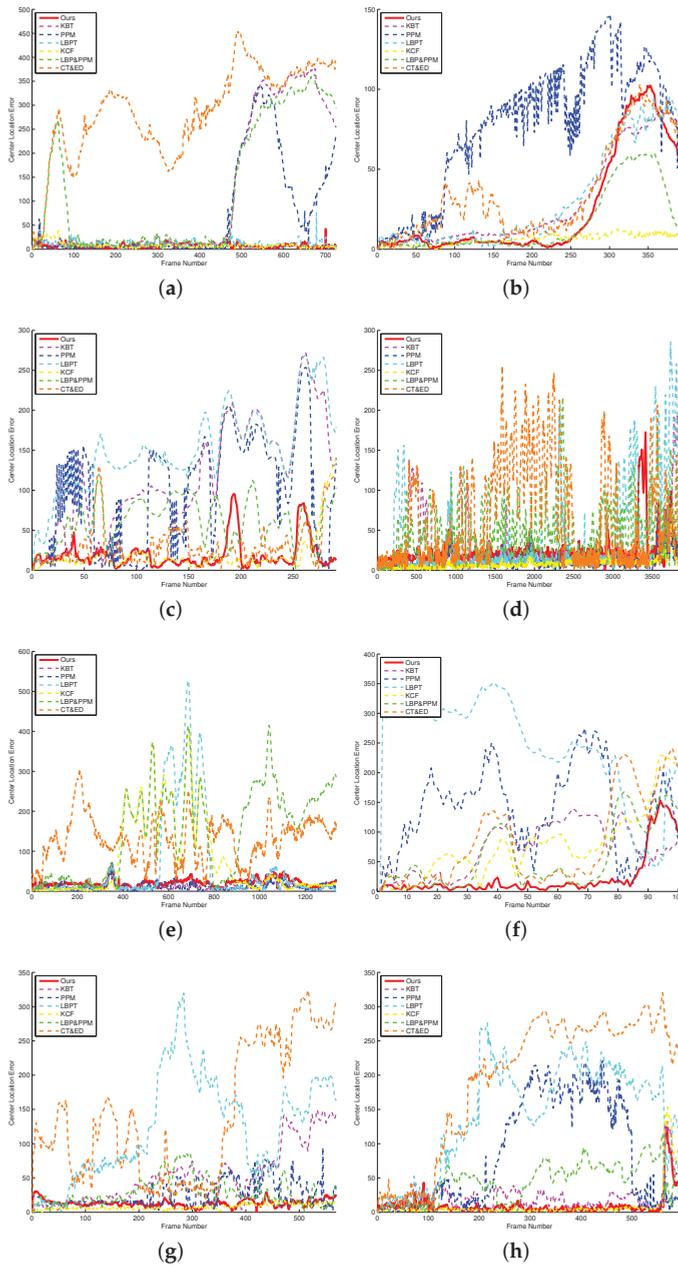


Figure 9. Frame-by-frame comparison of center location errors (in pixels) on eight challenging sequences. Based on the experimental results, our algorithm is able to track targets accurately and stably. (a) Basketball; (b) Car; (c) Coke; (d) Doll; (e) Lemming; (f) Matrix; (g) Trellis; (h) Woman.

4.4. Speed Analysis and Discussions

Table 6 lists the needed computation times of the five trackers on our test platform. The trackers run from 160 fps to 60 fps in the current Matlab implementation. The speed of the trackers depends on the area of the candidate region for all the test sequences and the number of iterations. Comparing with KBT, PPM, and KCF, LBPT and the proposed method spend lots of time on texture feature computation. However, they just calculate parts of useful points. Comparing with KBT, KCF and LBPT, PPM and our algorithm can calculate the target model and the search region by joint points to decrease the computational complexity. Because the dimension of the CT feature is 640 compared with KBT, PPM, LBPT, KCF, our tracker takes more time than the other trackers. However, the computational time can satisfy real-time applications.

Table 6. Computation speed comparison (fps).

SEQUENCE	KBT [10]	PPM [15]	LBPT [20]	KCF [9]	Proposed
Average success rate	164	100	88	165	66

5. Conclusions

A new object tracking method has been proposed in this paper. The algorithm can overcome some difficulties in real scenes such as object occlusion, sudden illumination changes, similarly colored backgrounds, and low illumination color images. This work integrates the outcomes of the color texture feature and PPM centroid iteration tracking. A color texture model called the CT feature is introduced. In addition, we propose using a posterior probability measure with the CT feature for target location. Three target model update strategies are designed to improve the tracking accuracy.

The tracking algorithm only using color can not track the target at similarly colored regions or low illumination regions. The combination of the color and the texture feature can overcome these difficulties, and the SDCS-LBP is a texture feature, which is robust against gray-scale changes. In real scenes, our algorithm shows a good performance. As our method is based on the histograms of the regions, it can overcome the problem of object partial occlusion. PPM measure and the target update strategies can reduce the tracking mistakes. In the experiments, our algorithm performs better than others for most of the test sequences. Future work will be dedicated to decreasing the complexity of the algorithm.

Acknowledgments: We would like to extend our sincere gratitude to our partner, Qing Zhou, for her careful revision and useful advice on this paper. This research is supported by the National Natural Science Foundation of China (Grant No. 61203350) and the Fundamental Research Funds for the Central Universities.

Author Contributions: Wenhua Guo developed the tracking algorithm and designed the experiments; Zuren Feng supervised the research and gave some useful advice for the tracker; and Xiaodong Ren reviewed the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Welch, G.; Bishop, G. SCAAT: Incremental Tracking with Incomplete Information. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 3–8 August 1997; ACM Press: New York, NY, USA; Addison-Wesley Publishing Co.: New York, NY, USA, 1997; pp. 333–344.
2. Isard, M.; Blake, A. Condensation—Conditional Density Propagation for Visual Tracking. *Int. J. Comput. Vis.* **1998**, *29*, 5–28.
3. Choo, K.; Fleet, D. People tracking using hybrid Monte Carlo filtering. In Proceedings of the Eighth IEEE International Conference on Computer Vision, Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 321–328.

4. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. In Proceedings of the IJCAI'81 Proceedings of the 7th international joint conference on Artificial intelligence, Vancouver, BC, Canada, 24–28 August 1981; Volume 81, pp. 674–679.
5. Reid, D. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control* **1979**, *24*, 843–854.
6. Cox, I.; Hingorani, S. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 138–150.
7. Comaniciu, D.; Ramesh, V.; Meer, P. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–577.
8. Jepson, A.; Fleet, D.; El-Maraghi, T. Robust online appearance models for visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 1296–1311.
9. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596.
10. Comaniciu, D.; Ramesh, V.; Meer, P. Real-time tracking of non-rigid objects using mean shift. In Proceedings of the Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, 15 June 2000; Volume 2, pp. 142–149.
11. Cai, Y.; Freitas, N.D.; Little, J.J. Robust Visual Tracking for Multiple Targets. In *Computer Vision—ECCV 2006*; Number 3954 in Lecture Notes in Computer Science; Leonardis, A., Bischof, H., Pinz, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 107–118.
12. Kim, D.; Kim, H.; Lee, S.; Park, W.; Ko, S. Kernel-Based Structural Binary Pattern Tracking. *IEEE Trans. Circ. Syst. Video Technol.* **2014**, *24*, 1288–1300.
13. Joukhadar, A.; Scheuer, A.; Laugier, C. Fast contact detection between moving deformable polyhedra. In Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyongju, Korea, 17–21 October 1999; Volume 3.
14. Liu, T.L.; Chen, H.T. Real-time tracking using trust-region methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 397–402.
15. Feng, Z.; Lu, N.; Jiang, P. Posterior probability measure for image matching. *Pattern Recognit.* **2008**, *41*, 2422–2433.
16. Haritaoglu, I.; Flickner, M. Detection and tracking of shopping groups in stores. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; Volume 1.
17. Heikkila, M.; Pietikainen, M. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 657–662.
18. Collins, R.T.; Liu, Y.; Leordeanu, M. Online selection of discriminative tracking features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1631–1643.
19. Wang, J.; Yagi, Y. Integrating color and shape-texture features for adaptive real-time object tracking. *IEEE Trans. Image Process.* **2008**, *17*, 235–240.
20. Ning, J.; Zhang, L.; Zhang, D.; Wu, C. Robust object tracking using joint color-texture histogram. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 1245–1263.
21. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
22. Ke, Y.; Sukthankar, R. PCA-SIFT: A more distinctive representation for local image descriptors. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004; Volume 2.
23. Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615–1630.
24. Bay, H.; Tuytelaars, T.; van Gool, L. Surf: Speeded up robust features. In *Computer Vision—ECCV 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
25. Alahi, A.; Ortiz, R.; Vandergheynst, P. Freak: Fast retina keypoint. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 510–517.
26. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987.
27. Ahonen, T.; Hadid, A.; Pietikainen, M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041.

28. Heikkilä, M.; Pietikäinen, M.; Schmid, C. Description of interest regions with local binary patterns. *Pattern Recognit.* **2009**, *42*, 425–436.
29. Tan, X.; Triggs, B. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Trans. Image Process.* **2010**, *19*, 1635–1650.
30. Wu, Y.; Lim, J.; Yang, M.H. Online Object Tracking: A Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013.
31. Wang, N.; Wang, J.; Yeung, D.Y. Online Robust Non-negative Dictionary Learning for Visual Tracking. In Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013; pp. 657–664.
32. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Strong Tracking Spherical Simplex-Radial Cubature Kalman Filter for Maneuvering Target Tracking

Hua Liu and Wen Wu *

Ministerial Key Laboratory of JGMT, Nanjing University of Science and Technology, Nanjing 210094, China; peterliuh@126.com

* Correspondence: wuwen@njjust.edu.cn; Tel.: +86-139-5102-5028

Academic Editors: Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang

Received: 20 February 2017; Accepted: 28 March 2017; Published: 31 March 2017

Abstract: Conventional spherical simplex-radial cubature Kalman filter (SSRCKF) for maneuvering target tracking may decline in accuracy and even diverge when a target makes abrupt state changes. To overcome this problem, a novel algorithm named strong tracking spherical simplex-radial cubature Kalman filter (STSSRCKF) is proposed in this paper. The proposed algorithm uses the spherical simplex-radial (SSR) rule to obtain a higher accuracy than cubature Kalman filter (CKF) algorithm. Meanwhile, by introducing strong tracking filter (STF) into SSRCKF and modifying the predicted states' error covariance with a time-varying fading factor, the gain matrix is adjusted on line so that the robustness of the filter and the capability of dealing with uncertainty factors is improved. In this way, the proposed algorithm has the advantages of both STF's strong robustness and SSRCKF's high accuracy. Finally, a maneuvering target tracking problem with abrupt state changes is used to test the performance of the proposed filter. Simulation results show that the STSSRCKF algorithm can get better estimation accuracy and greater robustness for maneuvering target tracking.

Keywords: maneuvering target tracking; spherical simplex-radial rule; cubature Kalman filter; fading factor; strong tracking filter

1. Introduction

Maneuvering target tracking has drawn increasing attention because of its widespread application in areas such as radar tracking, aircrafts surveillance, and spacecraft orbit control [1,2]. For maneuvering target tracking, many algorithms are developed and grouped into two types. One type is to improve the accuracy of the motion model, such as multiple-model (MM) methods [3], optimization of multiple model neural filter [4], current statistical (CS) model [5,6], and so on. The other type is to detect the target maneuverability and then to cope with it effectively, such as strong tracking filter (STF) [7], tracking algorithm based on maneuvering detection [8], and so on. In these methods, the performance of the filter is an important factor affecting the performance of these methods. Therefore, improving the accuracy of the filter is also a useful method to improve the performance of maneuvering target tracking. Thus, a large number of nonlinear filters have been developed. Among these algorithms, the extended Kalman filter (EKF) [9] is one of the earliest and most widely used nonlinear filters. The EKF uses a linearization technique, based on the first-order Taylor series expansion, and approximates the nonlinear system. However, EKF has some limitations, such as complex Jacobian matrix calculations and poor accuracy in estimating the states of the strongly nonlinear system.

As better alternatives to the EKF, many nonlinear filters based on the idea of Bayesian theory have been proposed. One popular approach for the nonlinear non-Gaussian filtering problem is to use sequential Monte Carlo methods. The most famous method is known as particle filter (PF) [10–13]. The key idea of PF is to represent the posterior distribution by a set of random samples

and to calculate estimates based on these samples and weights. Although the PF can provide good performance, the computational cost is very high and suffers from the curse of the dimensionality problem. These shortcomings restrict their applications in a real-time system. A different approach for nonlinear filtering is based on the point-based filtering technique that approximates intractable integrals encountered by a set of deterministically sampled points. Compared with the Monte Carlo numerical integration that relies on randomly sampled points, the deterministic point-based method has lower computational complexity with high accuracy. The type of filter includes the unscented Kalman filter (UKF) [14], Gauss-Hermite filter (GHF) [15], central difference filter (CDF) [16], etc. Among these methods, the well-known filter is UKF. The UKF uses unscented transform (UT) to capture the mean and covariance of a Gaussian density. It is shown that the UKF has better performance than the EKF. Besides its higher approximation accuracy, this UKF can avoid the cumbersome evaluation of Jacobian and Hessian matrices, making the algorithm easier to implement. Nevertheless, the unscented transform of the UKF is potentially unstable [17], which restricts its practical applications. Apart from the aforementioned filters, the cubature Kalman filter (CKF) has been proposed [17,18] by Arasaratnam and Haykin. Making use of the third-degree spherical-radial cubature rule, the CKF is reported to be more flexible in implementation form and more stable than UKF. In addition, Jia et al. [19] proposed the high-degree CKF where the number of sample points increases rapidly with the increase of the degree or state dimension. To further improve estimation accuracy with low complexity, a new nonlinear filter named spherical simplex-radial cubature filter (SSRCKF) is developed in [20]. The new class of CKF is based on the simplex spherical radial (SSR) rule, which improves the accuracy of CKF with only two more cubature points necessary.

Although the SSRCKF can achieve good accuracy in tracking non-maneuvering or weak maneuvering targets, it may lose the tracking ability to the abrupt state change when the system reaches the stable state. This is because the reaction of the gain matrix is delayed to the sudden change of the prediction error. To tackle the problem mentioned above, a new algorithm called strong tracking spherical simplex-radial cubature Kalman filter (STSSRCKF) is proposed in this paper. The STSSRCKF is developed based on the combination of strong tracking filter (STF) [7,21,22] and SSRCKF. The new algorithm using the strong tracking idea and the fading factor based on the residual to modify the prior covariance matrix quickly. Thus, the gain of the filter can be adjusted in real time to enhance tracking capacity for the maneuvering target. In addition, the algorithm can also keep a normal tracking accuracy for weak maneuvering targets. Compared with the STF, strong tracking unscented Kalman filter (STUKF) [23], strong tracking cubature Kalman filter (STCKF) [24] and SSRCKF, the proposed algorithm has a good accuracy and robust advantage over a wide range of maneuver. The performance of the proposed filter is demonstrated by the simulation.

The remainder of this paper is organized as follows. The overview of the background theory is presented in Section 2. The proposed algorithm is developed in Section 3. Simulation results and performance comparisons are presented in Section 4. Finally, conclusions are provided in Section 5.

2. A Review of UKF and CKF

The nonlinear discrete-time system is represented by

$$\begin{cases} \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \end{cases} \quad (1)$$

where $k \in N$ denotes discrete time, $\mathbf{f}(\cdot)$ represents the nonlinear function, $\mathbf{h}(\cdot)$ represents the measurement function. $\mathbf{x}_k \in \mathbf{R}^n$ is the state vector of system, $\mathbf{z}_k \in \mathbf{R}^m$ is the measurement, $\mathbf{w}_k \in \mathbf{R}^n$ is the process noise vector, and $\mathbf{v}_k \in \mathbf{R}^m$ is the measurement noise vector. The process \mathbf{w}_k and measurement noise \mathbf{v}_k are uncorrelated zero-mean Gaussian white sequences and have zero cross-correlation with each other, represented as $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$ and $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$, respectively.

Under the Gaussian assumption in the Bayesian filtering framework, the key problem of the nonlinear filtering problem is to calculate the multi-dimensional integrals. However, in most cases,

the multi-dimensional integrals cannot be solved analytically. As a result, several approximation methods have been proposed, such as the unscented transformation (UT) and the cubature rule.

The UT with $2n + 1$ the sigma points χ_i and corresponding weights is chosen as

$$\left\{ \begin{array}{l} \chi_0 = \mathbf{x}_{k|k} \\ \chi_i = \mathbf{x}_{k|k} + \left[\sqrt{(n + \lambda) \mathbf{P}_{k|k}} \right]_i \quad (i = 1, \dots, n) \\ \chi_i = \mathbf{x}_{k|k} - \left[\sqrt{(n + \lambda) \mathbf{P}_{k|k}} \right]_i \quad (i = n + 1, \dots, 2n) \\ \omega_m^{(0)} = \lambda / (n + \lambda) \\ \omega_c^{(0)} = \lambda / (n + \lambda) + (1 - \alpha^2 + \beta) \\ \omega_m^{(i)} = \omega_c^{(i)} = 1 / 2(n + \lambda) \quad (i = 1, \dots, 2n) \end{array} \right. \quad (2)$$

where $[\mathbf{P}_{k|k}]_i$ is the i th column of the matrix square root of $\mathbf{P}_{k|k}$, n is the dimension of state. $\lambda = \alpha^2(n + \kappa) - n$ is the scaling parameter; α determines the spread of the sigma points around $\mathbf{x}_{k|k}$. The positive constants β and κ are used as parameters of the method.

The third-degree cubature rule with $2n$ cubature points and weights is given by:

$$\left\{ \begin{array}{l} \chi_i = \mathbf{x}_{k|k} + \left[\sqrt{(n/2) \mathbf{P}_{k|k}} \right]_i \quad (i = 1, \dots, n) \\ \chi_i = \mathbf{x}_{k|k} - \left[\sqrt{(n/2) \mathbf{P}_{k|k}} \right]_i \quad (i = n + 1, \dots, 2n) \\ \omega_m^{(i)} = \omega_c^{(i)} = 1 / 2n \quad (i = 1, \dots, 2n) \end{array} \right. \quad (3)$$

As indicated above, the main difference between the UT used in UKF and the third-degree cubature rule used in CKF is that the UT has one more point in the center with a tune parameter κ . If the parameter κ is set to zero, the sigma points set will evolve into the cubature points set and the UKF becomes identical to the CKF. For UKF, the scaling parameter κ is always set to $n - 3$. Based on this point, for high-dimensional problems ($n > 3$), it will lead to the negative weight of the center point. The presence of the negative weight may lead the covariance matrix to become non-positively defined. Thus, the cubature rule is more stable than the UT. In summary, the CKF is virtually a special case of UKF and the CKF has better numerical stability than UKF.

3. Strong Tracking Spherical Simplex-Radial Cubature Kalman Filter

The heart of the spherical simplex cubature Kalman filter is the spherical-radial cubature rule. The spherical-radial cubature rule does not approximate the nonlinear function, but it can approximate the integral of the form (nonlinear function \times Gaussian) using weighted quadrature point sets. The integral with the standard Gaussian distribution $N(\mathbf{x}; 0, \mathbf{I})$ can be approximated by the quadrature

$$\int_{\mathbb{R}^n} f(\mathbf{x}) N(\mathbf{x}; 0, \mathbf{I}) d\mathbf{x} \approx \sum_{i=1}^m \omega_i f(\gamma_i) \quad (4)$$

where m is the total number of quadrature points in the state-space \mathbb{R}^n , $\{\gamma_i, \omega_i\}_i^m$ is a set of quadrature points and corresponding weights. The general Gaussian integral $\int_{\mathbb{R}^n} f(\mathbf{x}) N(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}) d\mathbf{x}$ can be approximated by the following transformation

$$\begin{aligned} \int_{\mathbb{R}^n} f(\mathbf{x}) N(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}) d\mathbf{x} &= \int_{\mathbb{R}^n} f(\sqrt{\mathbf{P}}\mathbf{x} + \hat{\mathbf{x}}) N(\mathbf{x}; 0, \mathbf{I}) d\mathbf{x} \\ &\approx \sum_{i=1}^m \omega_i f(\sqrt{\mathbf{P}}\gamma_i + \hat{\mathbf{x}}) \end{aligned} \quad (5)$$

The computational complexity of the numerical integration is proportional to the number of quadrature points, and the accuracy of the numerical integration rule is usually assessed by the polynomial approximation degrees.

3.1. Review of the Third-Degree Spherical Simplex-Radial Cubature Rule

The SSRCKF algorithm has the same structure as the general Gaussian approximation filters, such as the CKF, but uses the third-degree spherical simplex-radial cubature rule to calculate the Gaussian weight integral $I(f) = \int_{\mathbb{R}^n} f(x)N(x;0, I)dx$. By using the spherical simplex-radial cubature rule, the SSRCKF method can get more accurate estimation than CKF. In the third-degree spherical simplex-radial cubature rule, the following integral is considered [19]:

$$I(f) = \int_{\mathbb{R}^n} f(x) \exp(-x^T x) dx \tag{6}$$

where $f(\cdot)$ is arbitrary nonlinear function, \mathbb{R}^n is the integral domain. To calculate the above integral, let $x = rs$ ($s^T s = 1, r = \sqrt{x^T x}$). Equation (6) can be transformed into the spherical-radial coordinate system

$$I(f) = \int_0^\infty \int_{U_n} f(rs) r^{n-1} \exp(-r^2) d\sigma(s) dr \tag{7}$$

where $s = [s_1, s_2, \dots, s_n]^T, U_n = \{s \in \mathbb{R}^n : s_1^2 + s_2^2 + \dots + s_n^2 = 1\}$ is the spherical surface, and $\sigma(\cdot)$ is the area element on U_n . Then, the Equation (7) can be decomposed into the spherical integral $S(r) = \int_{U_n} f(rs) d\sigma(s)$ and the radial integral $I(f) = \int_0^\infty S(r) r^{n-1} \exp(-r^2) dr$.

3.1.1. Spherical Simplex Rule

As can be seen from the literature [25], the spherical integral $\int_{U_n} f(rs) d\sigma(s)$ can be approximated by the transformation group of the regular n -simplex with vertices a_j . P_0 The third-degree spherical simplex rule with $2n + 2$ quadrature points is given by

$$\begin{aligned} S(r) &= \frac{A_n}{2(n+1)} \sum_{j=1}^{n+1} (f(ra_j) + f(-ra_j)) \\ &= \sum_{j=1}^{N_s} \omega_{s,j} f(ry_j) \end{aligned} \tag{8}$$

where $A_n = 2\sqrt{\pi^n} / \Gamma^n(1/2), N_s = 2n + 2$.

3.1.2. Radial Rule

The radial integral $\int_0^\infty S(r) r^{n-1} \exp(-r^2) dr$ can be calculated by the following moment matching equation

$$\int_0^\infty S(r) r^{n-1} \exp(-r^2) dr = \sum_{i=1}^{N_r} \omega_{r,i} S(r_i) \tag{9}$$

where $S(r) = r^l$ is a monomial in r , with l an even integer. Using the moment method with the minimum number of points, the third-degree radial rule ($N_r = 1$) can be derived. From Equation (9) we can obtain the moments' equations as

$$\begin{cases} \omega_{r,1} r_1^0 = \frac{1}{2} \Gamma(\frac{n}{2}) \\ \omega_{r,1} r_1^2 = \frac{1}{2} \Gamma(\frac{n+2}{2}) = \frac{n}{4} \Gamma(\frac{n}{2}) \end{cases} \tag{10}$$

By solving Equation (10), the points and weights for the third-degree radial rule are given by

$$\begin{cases} r_1 = \sqrt{n/2} \\ \omega_{r,1} = \Gamma(n/2)/2 \end{cases} \tag{11}$$

3.1.3. Spherical Simplex-Radial Rule

By using Equations (7), (8) and (11), the third-degree spherical simplex-cubature rule ($N_r = 1$, $N_s = 2n + 2$) is given by

$$\begin{aligned} & \int_{\mathbb{R}^n} f(\mathbf{x})N(\mathbf{x}, 0, \mathbf{I})d\mathbf{x} \\ &= \frac{1}{\sqrt{\pi^n}} \int_{\mathbb{R}^n} f(\sqrt{2}\mathbf{x}) \exp(-\mathbf{x}^T \mathbf{x})d\mathbf{x} \\ &\approx \frac{1}{\sqrt{\pi^n}} \sum_{i=1}^{N_r} \sum_{j=1}^{N_s} \omega_{r,i} \omega_{s,j} f(\sqrt{2}r_i \mathbf{s}_j) \\ &= \frac{1}{2(n+1)} \times \left(\sum_{j=1}^{n+1} f(\sqrt{n}a_j) + \sum_{j=n+2}^{2n+2} f(-\sqrt{n}a_j) \right) \\ &= \sum_{k=1}^m \omega_k f(\xi_k) \end{aligned} \tag{12}$$

where $m = 2n + 2$, $\xi_k = \sqrt{n}[\mathbf{a}, -\mathbf{a}]_k$ and $\omega_k = 1/(2n + 2)$ are the corresponding weights.

The steps of SSRCKF algorithm for the nonlinear system can be found in the literature [17].

3.2. Strong Tracking Filter

To improve the performance of EKF, a concept of STF was proposed by Zhou and Frank [7]. They proved that a filter can obtain the strong tracking estimation of the state can have the strong tracking performance only if the filter satisfies the orthogonal principle [7]. In strong tracking, the time-varying suboptimal fading factor is incorporated, which online adjusts the covariance of the predicted state. In this way, the algorithm has the ability to track abrupt state change and strong robustness against mode uncertainties. The algorithm has the following steps [21]:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}_k(\hat{\mathbf{x}}_{k-1|k-1}) \\ \mathbf{P}_{k|k-1} &= \lambda_k \mathbf{F}_{k|k-1} \mathbf{P}_{k-1} \mathbf{F}_{k|k-1}^T + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})) \\ \mathbf{P}_{k|k} &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_{k|k-1} \end{aligned} \tag{13}$$

where $\mathbf{F}_{k|k-1}$ and \mathbf{H}_k are the process matrix and measure matrix, respectively. The suboptimal time-varying fading factor λ_k is given by

$$\lambda_k = \begin{cases} c_k, & c_k \geq 1 \\ 1, & c_k < 1 \end{cases}, c_k = \frac{\text{tr}[\mathbf{N}_k]}{\text{tr}[\mathbf{M}_k]} \tag{14}$$

$$\mathbf{N}_k = \mathbf{V}_k - \mathbf{H}_k \mathbf{Q}_{k-1} \mathbf{H}_k^T - \beta \mathbf{R}_k \tag{15}$$

$$\mathbf{M}_k = \mathbf{H}_k \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T \mathbf{H}_k^T \tag{16}$$

$$V_k = \begin{cases} v_0 v_0^T & k = 0 \\ \frac{\rho V_{k-1} + v_k v_k^T}{1 + \rho} & k \geq 1 \end{cases} \quad (17)$$

where $tr[\cdot]$ is the trace operation, $v_k = z_k - \hat{z}_{k|k-1}$ denotes the measurement residual vector; $\beta \geq 1$ is the softening factor, which can improve the smoothness of state estimation; $0 < \rho \leq 1$ is the forgetting factor. In generally, the parameters β and ρ are chosen as 4.5 and 0.95, respectively [26,27].

3.3. Equivalent Expression of the Fading Factor

As we know, STF need calculate the linearization of the nonlinear measurement matrix (Hessian matrix). However, SSRCKF is not necessary to compute the Hessian matrix. So we give the equivalent expression of STF, which need not calculate the Hessian matrix. Suppose $P_{k|k-1}^l$ is the state error covariance matrix before introducing fading factor, $P_{zz,k|k-1}^l$ is the measurement covariance matrix and $P_{xz,k|k-1}^l$ is cross-covariance matrix, Equations (15) and (16) have the following equivalent expressions:

$$N_k = V_k - (P_{xz,k|k-1}^l)^T (P_{k|k-1}^l)^{-1} Q_{k-1} (P_{k|k-1}^l)^{-1} (P_{xz,k|k-1}^l) - \beta R_k \quad (18)$$

$$M_k = P_{zz,k|k-1}^l - V_k + N_k + (\beta - 1) R_k \quad (19)$$

The new fading factor can be obtained through Equations (14) and (17)–(19). It can be verified from Equations (18) and (19) that the calculation of suboptimal fading factor in the Equation expression does not need to compute any Jacobian matrix.

3.4. Steps of the STSSRCKF

Based on the previous sections, the strong tracking spherical-simplex cubature Kalman filtering (STSSRCKF) can adjust the prediction error covariance matrix by introducing a suboptimal factor. Hence, the robustness and real-time tracking ability are provided in the STSSRCKF algorithm. The initial state is assumed to be Gaussian distribution with $\hat{x}_{0|0}$ and $P_{0|0}$. The computation steps of the third-degree strong tracking spherical simplex-radial cubature Kalman filter is summarized as follows:

Step 1. Give the state estimate $\hat{x}_{k-1|k-1}$ and the error covariance matrix $P_{k-1|k-1}$;

Step 2. State estimate prediction:

The cubature points are obtained as

$$\chi_{j,k|k-1}^l = \hat{x}_{k|k-1} + \text{chol}(P_{k|k-1})^T \zeta_j \quad (20)$$

where $\text{chol}(\cdot)$ is the Cholesky factorization.

Propagate the cubature points, the predicted state $x_{k|k-1}$, and the predicted covariance $P_{k|k-1}^l$ without the fading factor are given as

$$\chi_{j,k|k-1}^* = f(\chi_{j,k-1}) \quad (21)$$

$$\hat{x}_{k|k-1} = \sum_{j=1}^m \omega_j \chi_{j,k|k-1}^* \quad (22)$$

$$P_{k|k-1}^l = \sum_{j=1}^m \omega_j (\chi_{j,k|k-1}^* - \hat{x}_{k|k-1})(\chi_{j,k|k-1}^* - \hat{x}_{k|k-1})^T + Q_{k-1} \quad (23)$$

where Q_{k-1} is the covariance matrix of process noise.

Step 3. Calculation of the fading factor λ_k :

Using the predicted state $\hat{\mathbf{x}}_{k|k-1}$ and the predicted covariance $\mathbf{P}'_{k|k-1}$, the innovation covariance $\mathbf{P}'_{zz,k|k-1}$ and the cross covariance $\mathbf{P}'_{xz,k|k-1}$ can be calculated as

$$\mathbf{z}'_{j,k|k-1} = \mathbf{h}(\chi_{j,k|k-1}^*) \quad (24)$$

$$\hat{\mathbf{z}}'_{k|k-1} = \sum_{j=1}^m \omega_j \mathbf{z}'_{j,k|k-1} \quad (25)$$

$$\mathbf{P}'_{xz,k|k-1} = \sum_{j=1}^m \omega_j (\chi_{j,k|k-1}^j - \hat{\mathbf{x}}_{k|k-1}) (\mathbf{z}'_{j,k|k-1} - \hat{\mathbf{z}}'_{k|k-1})^T \quad (26)$$

$$\mathbf{P}'_{zz,k|k-1} = \sum_{j=1}^m \omega_j (\mathbf{z}'_{j,k|k-1} - \hat{\mathbf{z}}'_{k|k-1}) (\mathbf{z}'_{j,k|k-1} - \hat{\mathbf{z}}'_{k|k-1})^T + \mathbf{R}_k \quad (27)$$

The fading factor λ_k can be calculated by using Equations (14) and (17)–(19).

Step 4. Measurement updating modified by the fading factor:

The modified prediction covariance $\mathbf{P}'_{k|k-1}$ can be updated by

$$\mathbf{P}'_{k|k-1} = \lambda_k (\mathbf{P}'_{k|k-1} - \mathbf{Q}_{k-1}) + \mathbf{Q}_{k-1} \quad (28)$$

By utilizing the predicted state estimate $\hat{\mathbf{x}}_{k|k-1}$ and the modified predicted covariance $\mathbf{P}'_{k|k-1}$ with the fading factor λ_k , the modified predicted measurement $\hat{\mathbf{z}}'_{k|k-1}$, the modified cross covariance and the modified innovation covariance $\mathbf{P}'_{zz,k|k-1}$ can be calculated as follows

$$\chi'_{j,k|k-1} = \text{chol}(\mathbf{P}'_{k|k-1}) \xi_j + \hat{\mathbf{x}}_{k|k-1} \quad (29)$$

$$\mathbf{z}'_{j,k|k-1} = \mathbf{h}(\chi'_{j,k|k-1}) \quad (30)$$

$$\hat{\mathbf{z}}'_{k|k-1} = \sum_{j=1}^m \omega_j \mathbf{z}'_{j,k|k-1} \quad (31)$$

$$\mathbf{P}'_{xz,k|k-1} = \sum_{j=1}^m \omega_j (\chi'_{j,k|k-1} - \hat{\mathbf{x}}_{k|k-1}) (\mathbf{z}'_{j,k|k-1} - \hat{\mathbf{z}}'_{k|k-1})^T \quad (32)$$

$$\mathbf{P}'_{zz,k|k-1} = \sum_{j=1}^m \omega_j (\mathbf{z}'_{j,k|k-1} - \hat{\mathbf{z}}'_{k|k-1}) (\mathbf{z}'_{j,k|k-1} - \hat{\mathbf{z}}'_{k|k-1})^T + \mathbf{R}_k \quad (33)$$

Step 5. Estimation results:

The state estimate $\hat{\mathbf{x}}_k$ and the covariance \mathbf{P}_k at time k are calculate as follows

$$\mathbf{K}_k = \mathbf{P}'_{xz,k|k-1} (\mathbf{P}'_{zz,k|k-1})^{-1} \quad (34)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}'_{k|k-1}) \quad (35)$$

$$\mathbf{P}_k = \mathbf{P}'_{k|k-1} - \mathbf{K}_k \mathbf{P}'_{zz,k|k-1} \mathbf{K}_k^T \quad (36)$$

The STSSRCKF combines the advantages of STF and SSRCKF. Then the STSSRCKF has strong robustness against model uncertainties and good real-time state tracking capability [28]. Moreover, the STSSRCKF algorithm eliminates the cumbersome evaluation of Jacobian/Hessian matrices, its numerical stability and estimated accuracy are significantly improved.

4. Simulation and Results

The effectiveness of the proposed algorithm will be illustrated through two examples of maneuvering target tracking. Taking the root mean square error (RMSE) and accumulative RMSE (ARMSE), the study compared the STSSRCKF algorithm with the EKF algorithm and the SSRCKF algorithm.

4.1. Tracking Model and Measurement Model

The constant acceleration (CA) model is a common tool for tracking target modeling. The state Equation of CA model in two-dimensional case is described as follow:

$$\mathbf{X}_k = \text{diag}[\Phi_{CA}, \Phi_{CA}]\mathbf{X}_{k-1} + \mathbf{G}_{CA}\mathbf{V}_{k-1} + \mathbf{w}_{k-1} \quad (37)$$

where $\mathbf{X}_{k-1} = [x_{k-1}, \dot{x}_{k-1}, \ddot{x}_{k-1}, y_{k-1}, \dot{y}_{k-1}, \ddot{y}_{k-1}]^T$ is the target state at time $k-1$, (x_{k-1}, y_{k-1}) , $(\dot{x}_{k-1}, \dot{y}_{k-1})$ and $(\ddot{x}_{k-1}, \ddot{y}_{k-1})$ represent the target position, velocity and acceleration in the x and y coordinate at time $k-1$, respectively; $\text{diag}[\Phi_{CA}, \Phi_{CA}]$ is the state transition matrix, \mathbf{G}_{k-1} is the state input matrix, \mathbf{V}_{k-1} is the process noise, \mathbf{w}_{k-1} is zero-mean white Gaussian noise and its corresponding covariance matrix is \mathbf{Q}_{ca} . Φ_{CA} , \mathbf{G}_{k-1} are described as:

$$\Phi_{CA} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (38)$$

$$\mathbf{G}_{CA} = \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix} \quad (39)$$

where T is the sampling interval.

In radar tracking system, the target motion is usually modelled in Cartesian coordinates, whereas the target's position and azimuth are obtained in polar coordinate. The radar is located at the origin, and provides range and bearing measurements. The measurement model can be established as

$$\mathbf{z}_k = \begin{pmatrix} \sqrt{x_k^2 + y_k^2} \\ \text{atan2}(y_k, x_k) \end{pmatrix} + \mathbf{v}_k \quad (40)$$

where $\text{atan2}(\cdot)$ is the four-quadrant inverse tangent function, \mathbf{v}_k is the white Gaussian measurement noise with zero mean and covariance $\mathbf{R}_k = \text{diag}([\sigma_r^2, \sigma_\theta^2])$. σ_r and σ_θ denote the standard deviation of range measurement noise and bearing angle measurement noise, respectively.

4.2. Simulation of the STSSRCKF

Example 1. In this simulation, the sampling interval is $T = 1$ s and simulation time is 100s. The Monte Carlo simulations are carried out 200 times. The RMSE of the target position at time k and the accumulative RMSE (ARMSE) of estimated position at all times are defined in Equations (41) and (42):

$$\text{RMSE}_{\text{pos}}(k) = \sqrt{\frac{1}{M} \sum_{m=1}^M ((x_k - \hat{x}_{m,k})^2 + (y_k - \hat{y}_{m,k})^2)} \quad (41)$$

$$\text{ARMSE}_{\text{pos}} = \sqrt{\frac{1}{N} \sum_{k=1}^N (\text{RMSE}_{\text{pos}}^2(k))} \quad (42)$$

where M is the number of Monte Carlo runs, (x_k, y_k) is the actual value of the target position at time k and $(\hat{x}_{m,k}, \hat{y}_{m,k})$ is the estimated position at time k in m th Monte-Carlo. The RMSE and the accumulative RMSE in the velocity and acceleration can be defined in the same way.

This example considers a two-dimensional simulation scenario including one motion mode of high maneuver. The initial location of the target is $(x, y) = (100 \text{ m}, 400 \text{ m})$, its initial velocity is $(v_x, v_y) = (15 \text{ m/s}, 20 \text{ m/s})$, and its initial acceleration is $(a_x, a_y) = (0 \text{ m/s}^2, 0 \text{ m/s}^2)$. The target makes a uniform motion during the first 150 s. Then, it takes a high maneuver with the acceleration $(a_x, a_y) = (15 \text{ m/s}^2, 25 \text{ m/s}^2)$ up to the end of this simulation at $t = 200 \text{ s}$. In this simulation, the initial value $\hat{x}_{0|0}$ and the initial covariance matrix $P_{0|0}$ are set to be $[100 \text{ m}, 15 \text{ m/s}, 0 \text{ m/s}^2, 400 \text{ m}, 20 \text{ m/s}, 0 \text{ m/s}^2]^T$ and $\text{diag}[(50 \text{ m})^2, (20 \text{ m/s})^2, (1 \text{ m/s}^2)^2, (50 \text{ m})^2, (10 \text{ m/s})^2, (1 \text{ m/s}^2)^2]^T$, respectively. The standard deviation of range measurement noise σ_r is 30 m and the standard deviation of bearing angle measurement noise σ_θ is 10 mrad.

The example is executed to examine the performance among the SSRCKF, STF, STUKF, STCKF and STSSRCK methods. The RMSEs of the position, velocity and acceleration using the five filters are shown in Figures 1–3. It can be shown that the STF, STUKF, STCKF and STSSRCK methods can converge quickly when the target engages in high maneuvering. The SSRCKF algorithm only has a good performance for uniform motion. However, the performance of SSRCKF decreases seriously when the target engages in high maneuvering. This is because that the prediction covariance cannot be adjusted timely when the target state suddenly changes. The STF algorithm has the fourth speed of convergence, which is due to the fact that the linear approximation in the STF may introduce errors in the state which may lead the state to diverge. As can be seen from Figures 1–3, when the target is making uniform motion within the first 100 s, the five methods have a similar performance. When the maneuver starts at $t = 101 \text{ s}$, it obviously shows that STF, STUKF, STCKF and STSSRCKF have the ability to convergence. The main reason is that the fading factor can adjust the prediction covariance and the corresponding filter gain in real time, which makes these algorithms converge in a short time. We can also see that the RMSE of the proposed algorithm is lower than that of STUKF and STCKF. It means that estimate precision of the proposed algorithm is higher than that of the two algorithms. It is demonstrated that the proposed algorithm can effectively track the abrupt motion state of the target.

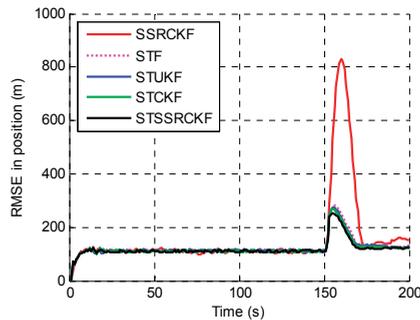


Figure 1. Root mean square error (RMSE) of the estimated position.

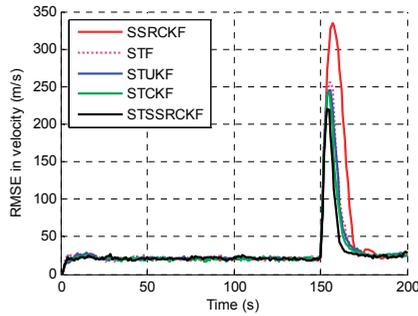


Figure 2. RMSE of the estimated velocity.

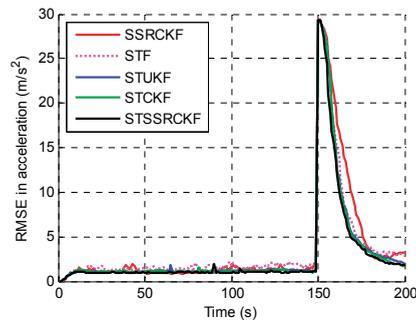


Figure 3. RMSE of the estimated acceleration.

To quantitatively describe the tracking performance, the ARMSEs of the five methods in estimating different target parameters are listed in Table 1. As shown, the STSSRCKF provided the best result in terms of estimation. The STCKF also performed well, followed by the STUKF and STF. The SSRCKF provided the worst estimate. We can also draw the conclusion that the STSSRCKF has the highest tracking accuracy of the position, velocity and acceleration.

Table 1. Tracking performance comparison.

Filters	Position ARMSE/m	Velocity ARMSE/(m/s)	Acceleration ARMSE/(m/s ²)
SSRCKF	152.1	31.2	6.9
STF	129.7	28.4	6.2
STUKF	124.5	27.5	5.9
STCKF	123.1	26.7	5.8
STSSRCKF	119.3	25.1	5.6

The program is made on the Intel Core (TM) i5-4430 3.0GHZ CPU with 4.00G RAM. Table 2 shows the computational complexity and the computational time of SSRCKF, STF, STUKF, STCKF and STSSRCK for each run. Apart from STF, the computational complexity of different filters is mainly determined by the number of points they use. The computational complexity of STCKF as well as STUKF differs only by one point. The computational complexity of SSRCKF and STSSRCKF is $O((2n + 2)^3)$, where n denotes the dimension of state. In addition, we can see that the computational complexity of STF is the lowest. Because there is a clear formula in STF to calculate the Jacobian matrix, the computational complexity of STF is much smaller than other four algorithms. It is also shown that the computational time of the SSRCKF is 0.07 s for each run. However, the computational time

of the STSSRCKF is 0.15 s, which is greater than that of the SSRCKF. This is because the STSSRCKF needs to calculate the suboptimal fading factor at each time step. At present, the time consumption is acceptable. The STSSRCKF needs more computational time than the SSRCKF, but considering the significant performance improvement gained from the STSSRCKF, this increased computational time is not substantial.

From this simulation, we can conclude that the STSSRCKF can perform the best in terms of the balance between computational complexity and estimation accuracy.

Table 2. Computational complexity and computational time of different filters.

Filters	Computational Complexity	Computational Time (s)
SSRCKF	$O\{(2n + 2)^3\}$	0.07
STF	$O\{n^2\}$	0.02
STUKF	$O\{(2n + 1)^3\}$	0.14
STCKF	$O\{(2n)^3\}$	0.14
STSSRCKF	$O\{(2n + 2)^3\}$	0.15

Example 2. This example evaluates the proposed algorithm in tracking a target with weak maneuver and medium maneuver. Therefore, two simulations are simulated as follows. Assume that there is a target making uniform at first. The initial location of the target is $(x, y) = (5000 \text{ m}, 5000 \text{ m})$, its initial velocity is $(v_x, v_y) = (150 \text{ m/s}, 80 \text{ m/s})$, and its initial acceleration is $(a_x, a_y) = (0 \text{ m/s}^2, 0 \text{ m/s}^2)$.

Case 1: Simulation of medium maneuvering target tracking. The target moves with initial acceleration until $t = 150 \text{ s}$. Then, it maneuvers with acceleration of $(a_x(151), a_y(151)) = (5 \text{ m/s}^2, 5 \text{ m/s}^2)$ up to end of this simulation at $t = 200 \text{ s}$.

Case 2: Simulation of weak maneuvering target tracking. The initial position, velocity and acceleration of the target are the same as those in Case1. The target also moves with initial acceleration until $t = 150 \text{ s}$. Then, it maneuvers with acceleration of $(a_x(151), a_y(151)) = (0.5 \text{ m/s}^2, 0.5 \text{ m/s}^2)$ up to end of this simulation at $t = 200 \text{ s}$.

Table 3 lists the accumulative RMSEs of the five methods in estimation the three target parameters. As can be seen from Table 3, the STSSRCKF algorithm also has a good tracking performance for a weak or medium maneuvering target.

Table 3. ARMSEs in simulation of medium and weak maneuvering target.

Simulation	Filters	Position ARMSE/m	Velocity ARMSE/(m/s)	Acceleration ARMSE/(m/s ²)
Case 1	SSRCKF	101.6	21.2	5
	STF	95.3	20.2	4.5
	STUKF	88.5	16.4	4.1
	STCKF	87.4	16.8	4.1
	STSSRCKF	81.1	15.9	3.7
Case 2	SSRCKF	50.5	8.2	1.8
	STF	65.1	10.8	2.4
	STUKF	57.3	8.8	2.2
	STCKF	56.3	8.3	2.2
	STSSRCKF	53.4	8.4	2.1

5. Conclusions

To implement higher tracking accuracy for a maneuvering target, a new method has been proposed based on the STF and SSRCKF algorithms. Firstly, the time-varying suboptimal fading factor is introduced in order to adjust the prediction covariance and the corresponding filter gain in

real time. Secondly, in the proposed method, the spherical simplex-cubature rule takes the place of calculating nonlinear function Jacobian matrix. In this way, STSSRCKF can converge rapidly in a short time. Thus, the proposed method has a high tracking accuracy for maneuvering target tracking. Simulation results show that the STSSRCKF can achieve higher accuracy and robustness than STF, STUKF, STCKF and SSRCKF, and indicate that it is suitable for maneuvering target tracking.

Author Contributions: All authors were involved in the study presented in this manuscript. Hua Liu provided insights into formulating the ideas and wrote the original manuscript of the paper. Wen Wu reviewed the manuscript and provided scientific advising to this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yepes, J.L.; Hwang, I.; Rotea, M. New algorithms for aircraft intent inference and trajectory prediction. *J. Guid. Control Dyn.* **2007**, *30*, 370–382. [CrossRef]
2. Liu, Z.X.; Zhang, Q.Q.; Li, L.Q.; Xie, W.X. Tracking multiple maneuvering targets using a sequential multiple target Bayes filter with jump Markov system models. *Neurocomputing* **2016**, *216*, 183–191. [CrossRef]
3. Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking. Part V: Multiple-model methods. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1255–1321.
4. Kazimiński, W.; Stateczny, A. Optimization of multiple model neural tracking filter for marine targets. In Proceedings of the 13th International Radar Symposium (IRS), Warsaw, Poland, 23–25 May 2012; pp. 543–548.
5. Zhou, H.R.; Kumar, K.S.P. A “Current” Statistical Model and Adaptive Algorithm for Estimating Maneuvering Targets. *AIAA J. Guid.* **1984**, *7*, 596–602.
6. Bingbing Jiang, W.S. Range tracking method based on adaptive “current” statistical model with velocity prediction. *Signal Process.* **2017**, 261–270. [CrossRef]
7. Zhou, D.H.; Frank, P.M. Strong tracking filtering of nonlinear time varying stochastic systems with coloured noise application to parameter estimation and empirical robustness analysis. *Int. J. Control* **1996**, *65*, 295–307. [CrossRef]
8. Li, X.R.; Jilkov, V.P. A Survey of Maneuvering Target Tracking—Part IV: Decision-Based Methods. In Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets 2002, Orlando, FL, USA, 15 April 2002; pp. 4728–4760.
9. Schmidt, S.F. The Kalman filter—Its recognition and development for aerospace applications. *J. Guid. Control Dyn.* **1981**, *4*, 4–7. [CrossRef]
10. Djuric, P.M.; Kotecha, J.H.; Zhang, J.Q.; Huang, Y.F.; Ghirmai, T.; Bugallo, M.F.; Miguez, J. Particle filtering. *IEEE Signal. Proc. Mag.* **2003**, *20*, 19–38. [CrossRef]
11. Doucet, A.; Johansen, A.M. A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. Available online: http://automatica.dei.unipd.it/tl_files/utenti/lucaschenato/Classes/PSC10_11/Tutorial_PF_doucet_johansen.pdf (accessed on 31 March 2017).
12. Martino, L.; Read, J.; Elvira, V.; Louzada, F. Cooperative parallel particle filters for online model selection and applications to urban mobility. *Digit. Signal. Process.* **2017**, *60*, 172–185. [CrossRef]
13. Kotecha, J.H.; Djuric, P.A. Gaussian particle filtering. *IEEE Trans. Signal. Process.* **2003**, *51*, 2592–2601. [CrossRef]
14. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [CrossRef]
15. Arasaratnam, I.; Haykin, S.; Elliott, R.J. Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature. *Proc. IEEE* **2007**, *95*, 953–977. [CrossRef]
16. Sadhu, S.; Srinivasan, M.; Bhaumik, S.; Ghoshal, I.K. Central difference formulation of risk-sensitive filter. *IEEE Signal. Process. Lett.* **2007**, *14*, 421–424. [CrossRef]
17. Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [CrossRef]
18. Arasaratnam, I.; Haykin, S. Cubature Kalman smoothers. *Automatica* **2011**, *47*, 2245–2250. [CrossRef]
19. Jia, B.; Xin, M.; Cheng, Y. High-degree cubature Kalman filter. *Automatica* **2013**, *49*, 510–518. [CrossRef]

20. Wang, S.Y.; Feng, J.C.; Tse, C.K. Spherical Simplex-Radial Cubature Kalman Filter. *IEEE Signal Process. Lett.* **2014**, *21*, 43–46. [CrossRef]
21. Wang, L.J.; Wu, L.F.; Guan, Y.; Wang, G.H. Online Sensor Fault Detection Based on an Improved Strong Tracking Filter. *Sensors* **2015**, *15*, 4578–4591. [CrossRef] [PubMed]
22. Ge, Q.B.; Shao, T.; Wen, C.L.; Sun, R.Y. Analysis on Strong Tracking Filtering for Linear Dynamic Systems. *Math. Probl. Eng.* **2015**, *2015*. [CrossRef]
23. Li, D.; Ouyang, J.; Li, H.Q.; Wan, J.F. State of charge estimation for LiMn2O4 power battery based on strong tracking sigma point Kalman filter. *J. Power Sources* **2015**, *279*, 439–449. [CrossRef]
24. Gao, M.; Zhang, H.; Zhou, Y.L.; Zhang, B.Q. Ground maneuvering target tracking based on the strong tracking and the cubature Kalman filter algorithms. *J. Electron. Imaging* **2016**, *25*, 023006. [CrossRef]
25. Jia, B.; Xin, M. Multiple sensor estimation using a new fifth-degree cubature information filter. *Trans. Inst. Meas. Control.* **2015**, *37*, 15–24. [CrossRef]
26. Jwo, D.J.; Wang, S.H. Adaptive fuzzy strong tracking extended kalman filtering for GPS navigation. *IEEE Sens. J.* **2007**, *7*, 778–789. [CrossRef]
27. Jwo, D.J.; Yang, C.F.; Chuang, C.H.; Lee, T.Y. Performance enhancement for ultra-tight GPS/INS integration using a fuzzy adaptive strong tracking unscented Kalman filter. *Nonlinear Dyn.* **2013**, *73*, 377–395. [CrossRef]
28. Han, P.; Mu, R.; Cui, N. Effective fault diagnosis based on strong tracking UKF. *Aircr. Eng. Aerosp. Technol.* **2011**, *83*, 275–282. [CrossRef]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Random Finite Set Based Bayesian Filtering with OpenCL in a Heterogeneous Platform

Biao Hu ^{1,†}, Uzair Sharif ^{1,†}, Rajat Koner ¹, Guang Chen ¹, Kai Huang ^{2,*}, Feihu Zhang ³,
Walter Stechele ⁴ and Alois Knoll ¹

¹ Robotics and Embedded Systems, Technische Universität München, 80333 München, Germany; hub@in.tum.de (B.H.); uzair.sharif@tum.de (U.S.); rajat.koner@tum.de (R.K.); guang@in.tum.de (G.C.); knoll@in.tum.de (A.K.)

² School of Data and Computer Science, Sun Yat-sen University, Xiaoguwei Island, Panyu District, Guangzhou 510006, China

³ School of Marine Science and Technology, Northwestern Polytechnical University, 127, Youyi West Road, Xi'an 710072, China; feihu.zhang@nwpu.edu.cn

⁴ Integrated Systems, Technische Universität München, 80333 München, Germany; walter.stechele@tum.de

* Correspondence: huangk36@mail.sysu.edu.cn; Tel.: +89-20-3244-2380

† These authors contributed equally to this work.

Academic Editors: Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang

Received: 27 February 2017; Accepted: 10 April 2017; Published: 12 April 2017

Abstract: While most filtering approaches based on random finite sets have focused on improving performance, in this paper, we argue that computation times are very important in order to enable real-time applications such as pedestrian detection. Towards this goal, this paper investigates the use of OpenCL to accelerate the computation of random finite set-based Bayesian filtering in a heterogeneous system. In detail, we developed an efficient and fully-functional pedestrian-tracking system implementation, which can run under real-time constraints, meanwhile offering decent tracking accuracy. An extensive evaluation analysis was carried out to ensure the fulfillment of sufficient accuracy requirements. This was followed by extensive profiling analysis to spot the potential bottlenecks in terms of execution performance, which were then targeted to come up with an OpenCL accelerated application. Video-throughput improvements from roughly 15 fps to 100 fps (6×) were observed on average while processing typical MOT benchmark videos. Moreover, the worst-case frame processing yielded an 18× advantage from nearly 2 fps to 36 fps, thereby comfortably meeting the real-time constraints. Our implementation is released as open-source code.

Keywords: random finite set Bayesian filtering; OpenCL; real-time execution

1. Introduction

Current success in the practical implementations of random finite set (RFS) filters has made it clear that RFS-based approaches are going to play a key role in the multisensor data fusion. This is mostly due to the probability hypothesis density (PHD) filters that present a recursive algorithm to jointly estimate target states in the presence of data association uncertainty, detection uncertainty, noise and false alarms [1]. Since then, the performance of new extensions has been increasing at a remarkable pace. The driving force behind the ever-increasing interest in RFS is its high potential in the applications of aerospace, robotics and intelligent systems, as presented in an excellent survey on the multisensor data fusion [2].

While the performance of RFS has been greatly improved, when dealing with real-world applications, running times become important. New computer technologies have already been proposed to accelerate the computation in many machine learning algorithms, but seldom applied

to data fusion algorithms. An emerging computing architecture that has been adopted in industry is the heterogeneous system architecture (HSA). HSA is a hardware platform that integrates central processing units, graphics processors or other processors (e.g., FPGA, DSP) on the same bus, with shared memory and tasks. With the support of the Open Computing Language (OpenCL), the HSA is able to relieve the programmer of the task of planning the moving of data between devices' disjoint memories, thus reducing the mutual communication latency and speeding up the computation.

In this paper, we investigate the problem of applying RFS filtering approaches to a heterogeneous platform, aiming to provide some insights on how to improve the RFS filtering running times by the heterogeneous system. For this purpose, we chose recently-proposed RFS-based filtering techniques i.e., PHD, labeled multi-Bernoulli (LMB) filters, to tackle the underlying multi-target tracking problem. PHD and LMB filters have been presented to be effective in tracking applications that require particle implementation or object individual existence probabilities. However, implementing the tracking algorithm in the heterogeneous system with OpenCL is a non-trivial problem. On the one hand, the tracking application by PHD or LMB filters is a complex and convoluted system that is intrinsically not suitable for parallel execution, as the parallel execution requires the system to be modular, and the execution within each module is independent. On the other hand, to efficiently use OpenCL to accelerate the execution, the execution bottlenecks should be spotted, and OpenCL configurations, such as running batch size or memory usage, should be well tuned. Last, the accuracy should be guaranteed while deploying OpenCL acceleration.

To this end, the tracking algorithms of this paper were developed in a highly modular system design approach and practically implemented in tracking pedestrians of a video. Specifically, we explored the use of the GM-PHD (Gaussian mixture-PHD) and SMC-LMB (sequential Monte Carlo-LMB) class of filters from this family, to implement the tracking algorithm. Initially, pure software (relying solely on CPU code in C++) implementations were carried out following a highly modular overall system-design approach. In order to parallelize the executions as much as possible, we made two major implementation modifications compared to the original GM-PHD or SMC-LMB. First, we did manual vectorization of code, as GM-PHD uses many small dimension matrices, which are not effective for large-scale parallelization. Second, we consider each Gaussian component independent of each other and splitting them into individual threads. Later, via extensive evaluation analysis concerning tracking accuracy, the GM-PHD trackers were found to be inadequate, and hence, the implementation focus was then shifted solely to SMC-LMB filters. To run the whole tracking application meeting real-time constraints, we carried out an extensive execution profiling of the algorithm. The spotted performance bottlenecks were then ported to the GPU accelerator using OpenCL programming constructs to perform the parallelization potential of the algorithm. We demonstrated the effectiveness of our developed approach by running the MOT benchmark [3]. In particular, we managed to improve from 15 fps to 100 fps in processing MOT video frames on average, while for the most computationally-expensive frame, we achieved 18× speedup improvements from 2 fps to 36 fps. We should note that our SMC-LBM implementation is based on the approach of [4] that separated the prediction and updating in the filtering. Now, another effective approach has been proposed to integrate the prediction and updating together so that the execution would be faster [5]. The discussion of them is in Section 4.

To the best of our knowledge, this is the first paper that reported the implementation results of the RFS-based filtering in a heterogeneous system with OpenCL. Moreover, since the implementation is designed in a highly modular way and the interface of each module is explicit, our developed software is able to be well integrated with other projects. The related open source code is released here (<https://github.com/nucleusbiao/Pedestrian-Tracking-using-SMC-LMB-with-OpenCL.git>).

This paper is structured as follows: Section 2 describes the mathematic model of GM-PHD filter, as well as the brief introduction of SMC-LMB. Section 3 presents the details of system implementation design. Section 4 presents the simulation and execution results. The paper is concluded in Section 5.

2. Preliminary

In the multi-target tracking problem, the number of targets to be tracked is unknown a priori and stochastically varies with time. At the sensor, a random number of measurements is received due to detection uncertainty and false alarms. Consequently, standard Bayesian filtering techniques are not directly applicable, since it is not known which of the received measurements, if any, should be used to update which target state, if any, at each sensor scan.

Towards this problem, the RFS approach is an emerging and promising alternative to the traditional association-based methods like joint probabilistic data association [6] and multiple hypothesis tracking filters [7]. Pioneered by Mahler [8,9], finite-set statistics analysis can be considered as the first systematic and rigorous approach to Bayesian state-estimation while explicitly avoiding the need for cumbersome association of measurements with targets or tracks. The RFS-based filtering enables the use of the optimal Bayesian estimation framework for multi-target tracking scenarios by introducing the concepts of a multi-target state/measurement expressed via random finite sets. In the following, we provide the mathematic background for the Gaussian-mixture PHD filter and briefly introduce the labeled multi-Bernoulli.

2.1. Gaussian Mixture-Probability Hypothesis Density Filter

In a multi-target tracking scenario, suppose that, at time t_{k-1} , there were $M(k-1)$ targets having states $(x_{k-1}^1, x_{k-1}^2, \dots, x_{k-1}^{M(k-1)})$ with each $x_{k-1}^i \in \mathcal{X}$. At t_k , some of these targets may die; the surviving targets evolve to their new states; and new targets may appear. This results in $M(k)$ targets having new states as $(x_k^1, x_k^2, \dots, x_k^{M(k)})$. Similarly, at the sensor, suppose that $N(k)$ measurements $(z_k^1, z_k^2, \dots, z_k^{N(k)})$ where each $z_k^i \in \mathcal{Z}$ are received at t_k . States and measurements at t_k can be aptly represented as finite sets, as shown below:

$$X_k = \{x_k^1, x_k^2, \dots, x_k^{M(k)}\} \in \mathcal{F}(\mathcal{X}) \tag{1}$$

$$Z_k = \{z_k^1, z_k^2, \dots, z_k^{N(k)}\} \in \mathcal{F}(\mathcal{Z}) \tag{2}$$

where $\mathcal{F}(\mathcal{X})$ and $\mathcal{F}(\mathcal{Z})$ denote the sets of all finite subsets of \mathcal{X} and \mathcal{Z} , respectively.

For a given multi-target state X_{k-1} at t_{k-1} , each $x_{k-1} \in X_{k-1}$ either continues to exist at t_k with survival probability $p_{S,k}(x_{k-1})$ or dies with probability $(1 - p_{S,k}(x_{k-1}))$. Consequently, for a given state x_{k-1} at t_{k-1} , its behavior at the next time step t_k is modeled as being an RFS $S_{k|k-1}(x_{k-1})$ that can take on either $\{x_k\}$ when the target survives or ϕ when the target dies. A new target born at t_k is similarly modeled by an RFS Γ_k . Using these quantities, we can express the multi-target state X_k at t_k to be:

$$X_k = \left(\bigcup_{\zeta \in X_{k-1}} S_{k|k-1}(\zeta) \right) \cup \Gamma_k \tag{3}$$

A given target $x_k \in X_k$ at t_k is either detected with detection probability $p_{D,k}(x_k)$ or missed with probability $(1 - p_{D,k}(x_k))$. Consequently, at each t_k , each state x_k generates a measurement RFS $\Theta_k(x_k)$ that can take on either $\{z_k\}$ when the target is detected, or ϕ when the target is missed by the sensor. In addition to these target-originated measurements, the sensor also receives a set of clutter measurements modeled via K_k RFS. Thus, given a multi-target state X_k at t_k , the multi-target measurement Z_k can be written as:

$$Z_k = \left(\bigcup_{x \in X_k} \Theta_k(x) \right) \cup K_k \tag{4}$$

In a similar fashion to STTBayes filtering, the multi-target state-transition densities $p(X_k|X_{k-1})$ and the multi-likelihood function $(p(Z_k|X_k))$ can be derived from the underlying physical models of targets and sensors using FISST techniques. Assuming their availability, the multi-target Bayes

filter propagates the multi-target posterior state-conditional density $p(X_k|Z^k)$ via the familiar prediction-update mechanism as follows:

$$p(X_k|Z^{k-1}) = \int p(X_k|X)p(X_{k-1}|Z^{k-1})\delta X \tag{5}$$

$$p(X_k|Z^k) = \frac{p(Z_k|X_k)p(X_k|Z^{k-1})}{\int p(Z_k|X_k)p(X_k|Z^{k-1})\delta X} \tag{6}$$

where the integrals in the recursion are FISST set integrals as introduced earlier.

To derive the GM-PHD recursion, the multi-target tracking context must also satisfy:

- Each target follows a linear Gaussian dynamical model, and the sensor has a linear Gaussian measurement model:

$$p(x|\zeta) = N(x; F_k\zeta, Q_k) \tag{7}$$

$$p(z|x) = N(z; H_kx, R_k) \tag{8}$$

- The survival and detection probabilities are state independent:

$$p_{S,k}(x) = p_{S,k} \tag{9}$$

$$p_{D,k}(x) = p_{D,k} \tag{10}$$

- The PHD or the intensity function of birth RFS γ_k is a Gaussian mixture (GM) of the form:

$$\gamma_k(x) = \sum_{i=1}^{J_{\Gamma,k}} w_{\Gamma,k}^i N(x; m_{\Gamma,k}^i, P_{\Gamma,k}^i) \tag{11}$$

Under these assumptions, it has been shown in [1] that the predicted posterior PHD, as well the posterior state PHD at any t_k is also a Gaussian mixture. Specifically, at t_k , if the prior PHD is expressed as a GM of the form:

$$v_{k-1}(x) = \sum_{i=1}^{J_{k-1}} w_{k-1}^i N(x; m_{k-1}^i, P_{k-1}^i) \tag{12}$$

then the GM-PHD recursion can be given by:

- Prediction:

$$v_{k|k-1}(x) = v_{S,k|k-1}(x) + \gamma_k(x) \tag{13}$$

$$v_{S,k|k-1}(x) = p_{S,k} \sum_{j=1}^{J_{k-1}} w_{k-1}^j N(x; m_{S,k|k-1}^j, P_{S,k|k-1}^j) \tag{14}$$

$$m_{S,k|k-1}^j = F_{k-1}m_{k-1}^j \tag{15}$$

$$P_{S,k|k-1}^j = Q_{k-1} + F_{k-1}P_{k-1}^jF_{k-1}^T \tag{16}$$

- Update:

$$v_k(x) = (1 - p_{D,k}(x))v_{k|k-1}(x) + \sum_{z \in Z_k} v_{D,k}(x; z)$$

$$v_{k|k-1}(x) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^i N(x; m_{k|k-1}^i, P_{k|k-1}^i) \tag{17}$$

$$v_{D,k}(x; z) = \sum_{j=1}^{J_{k|k-1}} w_k^j(z) N(x; m_{k|k}^j, P_{k|k}^j) \tag{18}$$

$$w_k^j(z) = \frac{p_{D,k}w_{k|k-1}^j q_k^j(z)}{\kappa_k(z) + p_{D,k} \sum_{l=1}^{J_{k|k-1}} w_{k|k-1}^l q_k^l(z)} \tag{19}$$

$$q_k^j(z) = N(z; H_k m_{k|k}^j, R_k + H_k P_{k|k-1}^j H_k^T) \quad (20)$$

$$m_{k|k}^j(z) = m_{k|k-1}^j + K_k^j(z - H_k m_{k|k-1}^j) \quad (21)$$

$$P_{k|k}^j = [I - K_k^j H_k] P_{k|k-1}^j \quad (22)$$

$$K_k^j = P_{k|k-1}^j H_k^T (H_k P_{k|k-1}^j H_k^T + R_k)^{-1} \quad (23)$$

As shown by these equations, the GM-PHD provides a computationally-efficient mechanism to propagate the posterior PHDs of multi-target state X_k ; though with the passage of time, the GM-PHD filter suffers from computation problems associated with the increasing number of such Gaussian components. In practice, this problem is dealt by carrying out special pruning procedures in a part of each recursion, which removes insignificant or negligible Gaussian components based on some pre-designed criterion.

2.2. The Labeled Multi-Bernoulli Filter

In addition to the GM-PHD filters, other RFS-based approximations of the multi-target Bayes filters include multi-Bernoulli filters and their various extensions. The idea of the multi-Bernoulli filter was first proposed by Mahler [10] proposing a novel multi-target multi-Bernoulli recursion as a tractable approximation to the recursive Bayes multi-target filter under low-clutter density scenarios, whereby a multi-Bernoulli RFS distribution propagates directly as an approximation to the posterior multi-target state X_k recursively instead of posterior PHDs.

However, [11] shows analytically that Mahler's multi-target multi-Bernoulli filter (MeMBeR) overestimates the cardinality and proposes a new variant called the cardinality balanced multi-target multi-Bernoulli (CBMeMBeR) filter. The CBMeMBeR filter extracts the cardinality bias of the MeMBeR filter in the update step and uses this to develop an unbiased update at the end of its recursion. Like the PHD/CPHD filters, [11] has provided closed-form GM-based solutions in the case of linear/Gaussian state-space models. For general non-linear/non-Gaussian considerations, SMC-based implementations have also been provided. Interested readers are encouraged to refer to [11] for a further conceptual understanding of multi-Bernoulli filters along with the detailed analysis of their prediction and filtering steps.

While multi-Bernoulli filters are not formulated to output target tracks, their generalization, referred to as the generalized labeled multi-Bernoulli (GLMB) filters [12], has been proposed to overcome this limitation. These filters rely on the notion of labeled RFSs for their working principles. We direct the readers to [12–14] for the large quantity mathematic formation of this approach. This paper mainly elaborates the implementation design.

3. System Design and Implementation

This section describes the main essence or the methodology in coming up with the overall design of the pedestrian tracking system. This is followed by a detailed overview of the techniques and strategies utilized in carrying out the implementation of the overall system.

3.1. System Design Modules

Like any good engineering design, the main focus has been to come up with a modular design approach to overcome the system complexity efficiently, while aiding in quick development of the system with each module being designed in an isolated fashion and having a clear notion of its input/output interfaces. Figure 1 presents a high-level abstracted view of the overall pedestrian tracking system resulting from this approach. We explain these modules briefly as follows.

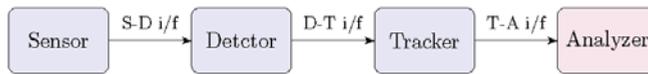


Figure 1. Block diagram of the multi-target tracking system.

3.1.1. Sensor

The sensor module represents an information-capturing device that extracts some useful target motion attributes (of interest) within the surveillance scene. This could be a stereo camera or a LiDAR sensor, etc., giving target motion information at regular intervals of time. Specifically for our work, we have made use of the following concepts as sensors:

- Simulated sensor model: For an extensive evaluation of the implemented pedestrian tracking system (Section 4), we design a simulation scenario simulating point targets whose motion follows linear/Gaussian characteristics. The specific simulation scenario can be considered as a form of sensor.
- Video frames: Likewise, we make use of MOT benchmark for further evaluation of the tracking system (Section 4). However, in this case, we are provided with camera video footage or frames comprising different kinds of pedestrian motions. These video frames then act as sensor observations.

3.1.2. Detector

The detector module is responsible for extracting the target-specific information from the sensor outputs. Generally, this involves coming up with target approximate kinematic quantities from the sensor scans to feed into the tracker module. In our work, the detector detects the individual target's 2D position coordinates within the surveillance region.

- Simulated detector model: For carrying out the simulation scenario, the simulated positions of the target from the sensor are corrupted with Gaussian noise to yield simulated detections. Furthermore, these detections are generated as part of the probabilistic process governed by a certain probability of detection p_d , which allows for target misdetections to help come up with robust tracker algorithms.
- Fast feature pyramid detector: The MOT benchmark provides detection annotations (2D position coordinates) on each of the training-video (sensor) frames. These detections are extracted by running the fast feature pyramid object detector algorithm, as proposed by Dollar et al. [15].
- Histogram of oriented gradients detector: The widely popular OpenCV library for developing computer-vision applications provides a working implementation of the HOG detector [16] targeting various platforms like C++, Python, CUDA, OpenCL, etc. We run this library function over MOT frames to come up with the target positions that are then fed into the tracker module further in the chain.

3.1.3. Tracker

The tracker module is the most crucial/significant processing element of the overall system as it is responsible for outputting the target/pedestrian tracks utilizing the detections within every sensor-scan. As will be explained later, we have employed mainly two trackers in our work:

- GM-PHD tracker: This tracker implements the GM-PHD filtering recursions to estimate the 4D (2D position, 2D velocity) target state. The original GM-PHD filtering algorithm [1] is enhanced using the techniques presented in [17] in order to be able to extract not just individual target states, but rather their trajectories or tracks.

- SMC-LMB tracker: This tracker carries out the implementation of the SMC/particle-filter-based LMB filter [12,13]. The LMB filtering is based on labeled RFSs, which helps to extract target tracks from their states automatically. The implementation is carried out in C++, as well as using OpenCL acceleration.

3.1.4. Analyzer

The analyzer module is the optional module responsible for analyzing the target tracks being produced from the tracker and compute various evaluation metrics enabling extensive evaluation of the implemented system (Section 4). Enabling this analysis, the analyzer module helps with coming up with stable efficient tracking system. Though, using this module could make sense in the development phase, it should be removed from the final system, as it provides no core functionality regarding pedestrian tracking.

3.1.5. System Design Interfaces

The main intra-module interfaces carried out in the project as shown in Figure 1 are briefly described below:

- Sensor input: Generally, sensor input consists of the whole surveillance view/region containing targets of interest. For our project, this represents either a simulated scenario or an MOT tracking scenario.
- Sensor-detector i/f: The interface between the sensor and the detector mainly represents the sensor outputs. Further processing tasks could be carried out as part of this interface for helping the detector in its algorithm. However, as part of this project, we simply forward the sensor output frames into the detector. The frames are structured as 2D pixel data embedding the target motion information.
- Detector-tracker i/f: This interface mainly represents the target detections, which act as input stimuli to the target algorithm. For our work, these detections are in the form of 2D position coordinates (i.e., in the form of a 2D floating-point vector).
- Tracker output: This primarily represents the overall output of the whole system. In this work, output involves individual target 4D states (a 4D floating-point vector) along with a specific label (a two-integer structure) being output from the tracker module within every sensor scan.

3.1.6. System Upgrades

As stated earlier, the proposed modular design methodology serves well to employ a plug and play-based design approach whereby one can easily replace an existing module for a better alternative without having to redesign the whole system from scratch, thus helping with efficient upgrading of the overall system. Some of the key upgrades in the present pedestrian tracking system could be:

- Sensor: There is no any specific requirement on the sensor. It can be a camera or a LiDAR. This would generate real-world sensor data for using the pedestrian tracking system in actual automotive scenarios to evaluate its effectiveness in carrying out its functionality.
- Detector: coming up with a detector algorithm of our own. This implemented detector would then help to do detections on the real scenario video footage.
- Tracker: further optimization of the implemented tracking algorithm, improving both the tracker accuracy, as well as its execution performance.

3.2. System Implementation

This subsection describes the major implementation aspect. As mentioned above, the sensor and detector are either simulated or are used from the MOT benchmark. The optional analyzer module will

be discussed at length in Section 4. Therefore, this subsection mainly focuses on the implementation of the tracker module.

We initially carried out the implementation of the GM-PHD algorithm, which lacked the ability to output target trajectories. This was overcome by using a tree-based approach to group the GM-terms of a single target together to provide a notion of its trajectory. After extensive simulation analysis (Section 4), we found the GM-PHD filter accuracy to be inadequate in dealing with general pedestrian tracking scenarios where the pedestrians deviated from linear/Gaussian motion characteristics. This led to the exploration of SMC approaches to offer better accuracy. In light of this, a particle-filter implementation for the LMB filter was successfully carried out. Later via extensive profiling, the LMB filter implementation in C++ was accelerated via OpenCL kernels by spotting the performance bottlenecks and re-implementing them using parallel programming constructs.

3.2.1. GM-PHD Tracker

The GM-PHD filter works by propagating the posterior PHD of the multi-target state in time during each of its recursions [1,17]. The GM-PHD filter recursion is carried out as shown in Figure 2. Each of the block represents a C++ class method performing its specific functionality. The arrows represent the data flow, whereby the GM terms representing the PHDs of multi-target state travel back and forth between the prediction and the update modules. Each of the Gaussian terms used throughout the filtering operation are compactly represented as a C++ struct with weight, mean and covariance as its attributes. To carry out the linear-algebra matrix operations in C++, we make use of efficient open-source library Armadillo. Therefore, while the weight is represented as a C++ float variable, the mean and covariance are better represented via armadilloclasses.

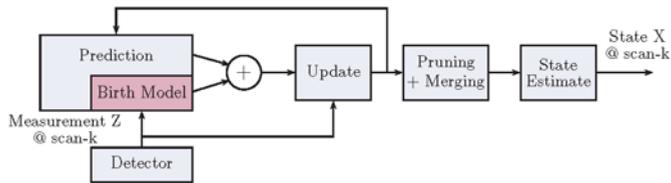


Figure 2. Block diagram of GM-probability hypothesis density (PHD) filter recursion.

At the start of each filter iteration/recursion, the new scan detections are used by the birth model, which compares them to the stored previous-scan detections. Based on likely association or similarities between a specific pair within these consecutive scans, the birth model forms new targets by assigning a new set of Gaussian terms, i.e., a Gaussian mixture as part of the predicted GM. These components are then mixed/added with the predicted GM of the surviving targets (surviving targets are represented by update GM in past iteration) obtained via (15) and (16).

Similarly in the update block, the current-scan detections are used along with the current computed prediction GM to extract the update GM using (21) and (22). These Update GM terms are then passed through the prune/merge block where we use a three-fold strategy to reduce the computational complexity arising from the increasing GM terms. These are summarized as follows:

- First, all of the close-by Gaussian terms (via their means) are merged together to form a composite Gaussian term, as they are thought to represent a single target.
- Then, all of the Gaussian terms whose weights are less than the filter-specified threshold are discarded as insignificant terms and are not processed further. This also allows one to gracefully terminate target tracks.
- Finally, we keep a cap on the maximum number of Gaussian terms corresponding to the maximum number of expected pedestrians within the surveillance zone.

After the pruning, the update GM represents the posterior PHD of the multi-target state with each Gaussian term representing a possible target state. To avoid tracking clutter terms, a second threshold is used here to discard the Gaussian terms that are not too significant as of the current iteration, but could lead to greater weights in coming iterations. Such terms are not output as current target states, but also not discarded, as they are being kept in the surviving target GM to be considered for the next iterations. For the significant terms, their means represent the individual target states and are output as such.

As should be obvious from these outputs, this preliminary filter is only capable of extracting individual target states, but does not output tracks or trajectories, i.e., there is no association between currently obtained states and the past ones. To overcome this limitation, the implementation algorithm is enhanced via tree-based techniques. For the sake of brevity, we ignore here the implementation details of those techniques, however. Interested readers are referred to [17] for further details in this regard.

3.2.2. SMC-LMB Tracker

The GM-PHD tracker provides the optimal PHD recursive solution in the case of targets and sensors following linear/Gaussian state-space characterizations. Therefore, expectedly, its accuracy performance should be deteriorated to a certain degree when the targets exhibit non-linear and/or non-Gaussian motion tendencies. We carry out simulation analysis (Section 4) to investigate this and find that the tracker performance is severely affected up to a point that the error is too much to tolerate. This led to the exploration of non-linear techniques to overcome this limitation.

SMC-based particle filters have been a popular approach in this context after being introduced in the 1990s. We looked into the possibility of using the particle filter implementation of the PHD filter in order to make it suitable for tackling generic pedestrian motions. However, we found that, recently, a new class of RFS-based filters called the LMB filters has been proposed, which are deemed to be more accurate than PHD filtering techniques. Furthermore, the SMC implementations of these filters match with those of PHD filters in terms of computational complexity [14]. Therefore, this work further focuses on implementing the SMC-LMB tracker as the main tracker module in Figure 1.

Figure 3 presents the overview of the implementation of the SMC-LMB tracker. Structurally, it is similar to Figure 2, like any Bayesian estimator, but computationally, there are major differences. First, instead of using GM to represent the posterior states, the SMC-LMB tracker relies on propagating a set of multi-Bernoulli terms. Similar to the GM-PHD filter implementation, these terms are compactly represented as a C++ struct using standard floats while the 4D state is conveniently represented via Armadillo vectors. Moreover, based on labeled RFSs, each instance of this struct has a unique 2D intlabel vector that acts as a tag for each tracked target. Each of the sub-blocks shown in Figure 3 is implemented via C++ functions as member functions of the LMB filter class.

- Filter initialization: This function is executed once for each instance of the tracker class at the time of its construction. Here, all of the tracker parameters are set, like the number of particles to represent state-pdfs, the maximum number of LMB components allowed to represent the posterior multi-target state, etc., along with the state-space modeling parameters.
- Birth model: Similar to GM-PHD filter implementation, the birth model in the SMC-LMB tracker implementation relies on the associations between the measurements obtained in consecutive scans. However, in the case of the birth of new targets, instead of representing it via a GM, a new multi-Bernoulli term is generated. In the current implementation, we find speeds in the Cartesian space between every detection pair using the current and the immediate past detection scan. If the speeds for a specific pair lie within the tracker-parametrized V_{max} value, the pair is deemed to correspond to a single target, and hence, a new target track is created. This track is initialized via parametrized existence probability while its state-pdf is supposed to be a Gaussian, and a certain number of particles are drawn from it stochastically. This number of particles is also parametrized, and we recommend them to be within powers of two for ease in GPU particle-level processing.

- Predict LMB: This module carries out the LMB prediction. Specifically, it generates predict-LMB terms for new-born targets from the birth model, as well as survive-LMB terms for existing targets via current update-LMB terms. We use two different sets of LMB terms instead of a single one, as it is much easier to do further conversion into δ -GLMB components separately and then merge them together.
- Predict-LMB to predict- δ -GLMB: Both of computed birth predict-LMB, as well as surviving predict-LMB terms are then converted to their δ -GLMB terms. This step is necessary to allow the δ -GLMB update later in the data flow. Now, even for a moderate number of targets/pedestrians, these δ -GLMB could become large, and processing them quickly becomes computationally expensive. Like for the case of the GM-PHD tracker, we introduce pruning schemes to cap the maximum number of components. However, in contrast to the former approach, the components have not yet been computed. Therefore, to avoid computation of all such components followed by the propagation of significant components, we rather formulate this problem as a K-shortest path problem and use the computationally-efficient Eppstein solution [18] (using the Bellman–Ford algorithm [19] internally) to directly compute only the significant components without the need for further pruning. Interested readers are encouraged to read [4] in order to come up with such a formulation.

After computing the separate δ -GLMB components for the new-born and existing targets, they are convolved together to give the overall δ -GLMB terms that are used for the update phase in the next scan/iteration while the LMB terms are simply concatenated together.

- Update LMB intermediate: This is the first step within the update step of the SMC-LMB recursion. It computes all possible update LMB terms based on every possible association of the current measurement-set with the previous scan predict LMB terms. These terms would be required later on in conversion of update δ -GLMB terms to their equivalent LMB terms. Hence, these terms are considered intermediate within the recursion.
- δ -GLMB update: This step performs the closed-form δ -GLMB update on the predict δ -GLMB [4] terms as obtained in the previous iteration. Here, again, we are confronted with the similar problems of rapidly growing terms and have to employ some sort of a cap on the maximum number of components to deal with computational complexity. However, because of the measurement involvement, this problem is formulated as the K-best assignment problem as opposed to the K-shortest path problem. To solve this problem, we rely on using the Murty algorithm [20] (using the Hungarian method internally) as explained in greater detail in [4].
- Update LMB: As shown in Figure 3, coming up with the update LMB terms within each tracker iteration involves a two-fold process. First, a conversion from update δ -GLMB terms to corresponding LMB terms is carried out such that the LMB set matches the PHD terms of the δ -GLMB set as was explained in [13]. Next, the particles needed to represent each of LMB term's state probability density functions $p^i(\cdot)$ are replaced with new set in a commonly-used procedure referred to as particle resampling to deal with the particle impoverishment problem. The computation of these LMB components completes the SMC-LMB recursion.
- Track management: In contrast to GM-PHD tracker, no special procedures are required to output target tracks, as the SMC-LMB filter outputs update LMB terms containing unique tags, i.e., outputting target tracks or trajectories directly. Here, again, the techniques of merging (to combine tracks formed from single target) and pruning (for tracks that we are not yet confident of being either a new target or clutter) are used just like for the case of the GM-PHD tracker.
- State estimation: The final step within each tracker iteration is to estimate individual target states and to associate them to already existing tracks/trajectories. For this, we use Mahler's ESFfunction [4] to first estimate stochastically the cardinality of the current multi-target state based on the pruned update LMB terms. Then, a certain number of most weighted/significant components corresponding to this cardinality estimate is chosen for state-estimation. Using the

particle representation of these components, an empirical measure is easily derived for each chosen component.

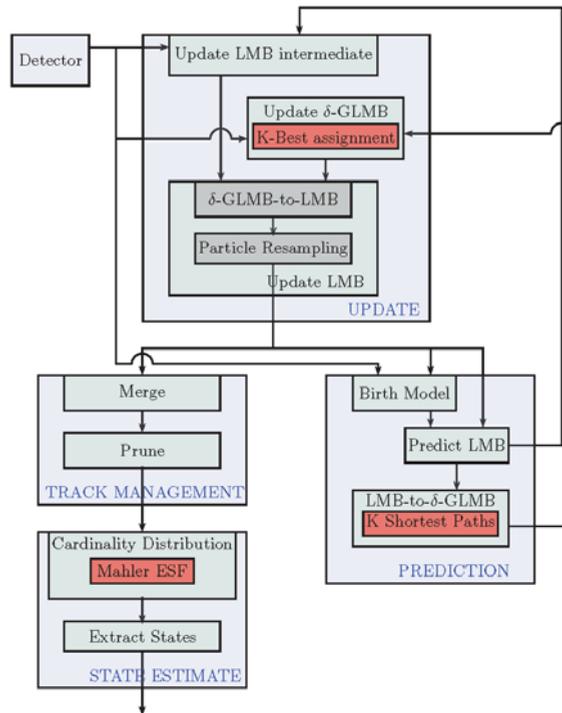


Figure 3. Block diagram of sequential Monte Carlo (SMC)-labeled multi-Bernoulli (LMB) tracker recursion.

3.3. OpenCL Acceleration

Being satisfied with the tracking accuracy of the SMC-LMB tracker (Section 4), an extensive profiling of the above-mentioned algorithm in C++ was carried out for the purpose of spotting potential performance bottlenecks. As clear from the detailed analysis presented in Section 4, the primary source of execution performance bottleneck within the recursion is the computation of update LMB terms, roughly amounting to 75% of the computations. Therefore, in our strategy to improve the execution performance of the algorithm, instead of redesigning the whole algorithm from scratch via programming constructs, we relied rather on a hybrid of C++ and OpenCL computation code. We transformed the sequential execution of the update LMB function into OpenCL kernels to significantly improve the timing performance of the whole algorithm. The main implementation aspects of this strategy are outlined below:

- Generation of uniform random numbers on the GPU itself using the AMD CLRNCompute library.
- Breaking down the for-loops within the update block down to the level of particle computations.
- Efficient parallel scan (prefix-sum) on the cumulative weight array to carry out the particle resampling procedure.
- Optimized memory organization for the LMB terms throughout the update part of the recursion as a high amount of memory transfers between the host CPU and GPU accelerator severely affects the performance and could possibly outdo the benefits achieved via GPU computations.

- The number of particles allocated for each of the LMB terms is chosen to be in powers of two, which makes it easier to use shared-memory optimizations within GPU computations for further acceleration of the application.
- We use the extensive vector operation module for vectorizing code.
- Each Gaussian component is computed in a separate thread; as the number of targets increases, the number of components increases very rapidly. This feature exploits all advantages of a parallel architecture.

4. Experimental Evaluations

This section presents a detailed evaluation and analysis of the pedestrian tracking system as explained in previous sections. Primarily, the analysis was carried out in two-fold. We discuss the tracking accuracy performance of the proposed design followed by its execution performance analysis.

4.1. Evaluation Metrics

It is paramount to have a clear notion of evaluation metrics before carrying out the actual evaluation of the system itself. Given below is a brief overview of these evaluation metrics, which will be used to evaluate the proposed system.

4.1.1. Tracking Accuracy

As mentioned earlier, the multi-target tracking problem attempts to jointly find the number of targets, as well as their individual states from the received measurements with the passage of time. Therefore, to quantify such a tracker's accuracy, we make use of following metrics:

- Cardinality estimate: this metric extracts the number of targets at the end of each tracker recursion. This can then be compared with the truth/actual cardinality of the multi-target state to figure out the tracking errors in this respect.
- Optimal sub-pattern assignment (OSPA): this metric defines a notion of mis-distance and corresponding error between actual and estimated individual target states as proposed firstly by [21].

4.1.2. Execution Performance

For profiling the tracker algorithm's execution performance, we use a simple mechanism involving the computation of the number of CPU cycles across the algorithm. In the context of video processing, we can obtain an average measure of frames per second as a computational throughput measure for the tracker via:

$$fps = \frac{\text{Number of frames in video}}{\text{Total execution time}} \quad (24)$$

4.2. Tracking Accuracy Analysis

This section details this work's findings regarding the tracking performance of the proposed LMB tracker. In order to evaluate the tracker for providing this analysis, we make use of a simulation analysis, as well as using available dataset videos.

4.2.1. GM-PHD and SMC-LMB Comparisons

For carrying out an extensive analysis for evaluating the tracking accuracy of the implemented LMB tracker, we created a 2D surveillance point-target-based simulation scenario. The target motion dynamics, as well as the sensor model were made to follow linear/Gaussian characteristics. Apart from that, the simulation was designed to be highly parameterized in the number of targets; their birth locations; their birth times and death times; their detection and survival probabilities; the amount of clutter; process/measurement noise, etc. This helps to generate a diverse range of

simulation scenarios to adequately understand the tracking accuracy performance under the influence of different constraints.

For the comparison, we designed two artificial simulation scenarios, whereby we simulated 12 linearly moving targets for a duration of 100 time steps. The first scenario was designed to show the accuracy performance of the algorithm under highly ideal tracking conditions, while the other scenario posed slightly more challenging conditions. These were parameterized as:

- Easy tracking scenario: $p_{S,k} = 0.98$, $p_{D,k} = 0.98$, $\sigma_v = 1 \text{ m/sec}^2$, $\sigma_w = 1 \text{ m}$, $\Delta = 1 \text{ s}$, $\lambda_c = 5$.
- Hard tracking scenario: $p_{S,k} = 0.90$, $p_{D,k} = 0.90$, $\sigma_v = 5 \text{ m/sec}^2$, $\sigma_w = 10 \text{ m}$, $\Delta = 1 \text{ s}$, $\lambda_c = 60$.

where $p_{S,k}$, $p_{D,k}$ are the survival and detection probabilities. Δ is the sensor sampling period or the inter scan-time. σ_v, σ_w represent the variances within the Gaussian process and measurement noises, respectively. The parameter λ_c denotes the clutter density, which is defined to be the average number of clutter returns.

As can be seen from Figure 4, the GM-PHD tracker performs apparently perfectly in the easy scenario for estimating the individual target states as illustrated by the OSPA measure. The cardinality estimate, however, shows that even in ideal conditions, the tracker does make occasional mistakes. This is attributed to the adaptive birth distribution model, being embedded inside the tracker algorithm, which requires some initial scans to confirm successive measurements of the new target in order to confirm it as a new track. This leads to cardinality errors at target births. Avoiding this can lead to tracker performance being severely affected, causing it to consider every new detection as a new target, which could very well be a clutter detection or a false-alarm. Similarly, when the target dies, the tracker expectedly makes mistakes because the algorithm cannot be sure about the target disappearance as an actual death or a miss-detection because of sensor imperfection. If one designs the tracker algorithm to abruptly terminate tracks just because of one misdetection, then tracker performance could suffer drastically. Consider for example the occurrence of a target misdetection. When such a target is re-detected, then the tracker would consider it to be a new target/track instead of continuing the previous known track. This behavior in most tracking scenarios is undesirable. Furthermore, these plots clearly show the worsening performance of the tracker in dealing with more severe tracking environments.

To evaluate the SMC-LMB tracker, we used a configuration of 512 particles and cap the update δ -GLMB hypotheses and the δ -GLMB components to a maximum of 100. Figure 5 presents the corresponding results under the designed simulation scenarios. It is a clear from these plots that using a sufficiently high number of particles/samples (like 512 in our case) to approximate the true posterior density, the LMB tracker shows comparable, if not better, performance than the more suitable GM-PHD tracker for linear/Gaussian systems. Especially under severe tracking scenarios where the targets show considerable deviation from the linear/Gaussian as governed by the higher σ_v value, these results justify the use of SMC-based trackers for MTT tracking, by showing their robustness in dealing with process modeling imperfections.

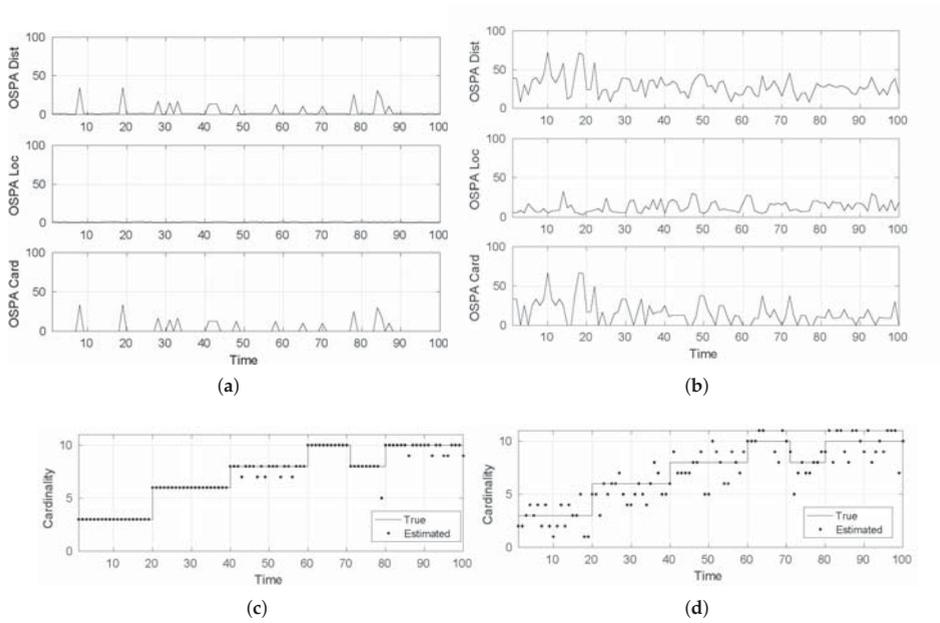


Figure 4. Tracking accuracy of the GM-PHD tracker. (a) Optimal sub-pattern assignment (OSPA) distance (easy-scenario); (b) OSPA distance (hard-scenario); (c) cardinality estimate (easy-scenario); (d) cardinality estimate (hard-scenario).

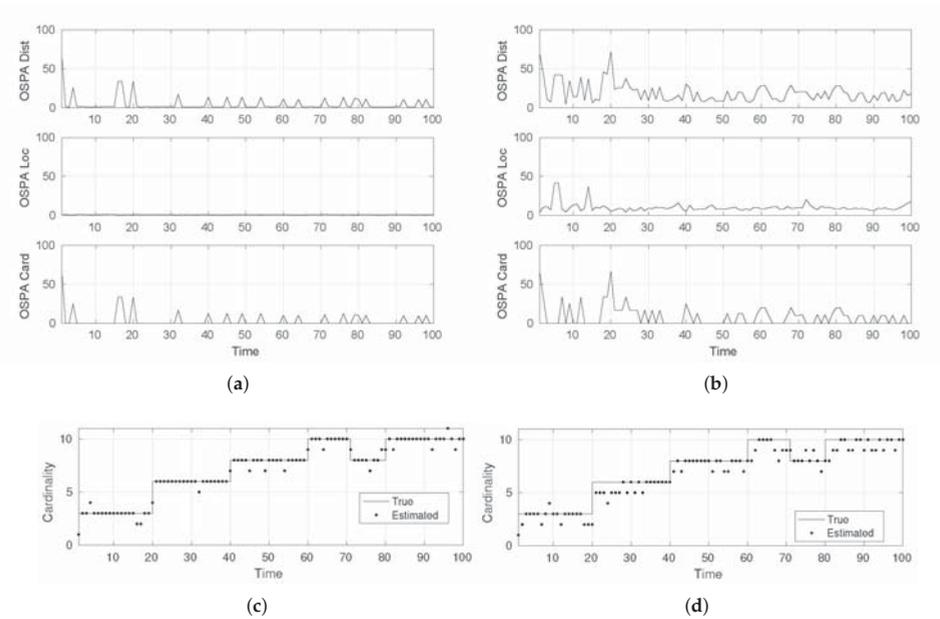


Figure 5. Tracking accuracy of the SMC-LMB tracker. (a) OSPA distance (easy-scenario); (b) OSPA distance (hard-scenario); (c) cardinality estimate (easy-scenario); (d) cardinality estimate (hard-scenario).

4.2.2. MOT Dataset Analysis

This subsection presents the tracking accuracy performance of the LMB tracker system on publicly available dataset videos. The goal of the analysis is to further build on the understanding of the tracking accuracy analysis developed via comprehensive simulations and to see whether the simulation results correspond to pedestrian tracking scenarios in actual video footage.

For this purpose, we use an effective benchmark named the Multiple Object Tracking Challenge [3]. This benchmark is designed to provide video footage covering diverse multi-target tracking contexts where the goal is to track pedestrians accurately. The benchmark is designed as a competition where the current state-of-the-art approaches are ranked as per their tracking accuracy. The pedestrian detections are already provided by the benchmark so the accuracy of tracking directly depicts the efficiency of the tracking algorithm in dealing with various tracking challenges.

For the sake of brevity, in this section, we present an evaluation analysis of the LMB tracker on two of the MOT videos as summarized in Table 1:

- KITTI-17: static camera; mostly linear target motion
- PETS09-S2L1 static camera; targets move in irregular patterns

Table 1. MOT dataset videos.

Video	Resolution	Number of Frames	Unique Targets	Maximum Targets per Frame	Target Density
KITTI-17	1224 × 370	145	9	4	4.7
PETS09-S2L1	768 × 576	795	19	8	5.6

Figure 6 shows the corresponding accuracy plots when these videos are fed to the LMB tracker for pedestrian tracking. For the KITTI-17 video, the tracker takes some initial frames to confirm target tracks, thereby making cardinality errors in the initial frames, but once the tracks are confirmed, the tracker performs highly accurately in tracking each of the pedestrian motions. The reason for such small OSPA distances in the later phase of the video can be attributed to the motion characteristics of the pedestrians. In the KITTI-17 video, most targets move in a straight line, i.e., in a linear fashion, hence the tracker performs as expected, as well as per the simulation analysis.

On the other hand, the PETS09-S2L1 video presents a much tougher challenge in that the pedestrians move in irregular patterns like moving abruptly or moving in circles, etc. As mentioned earlier, due to the use of linear state-space models internally in the current SMC-LMB tracker implementation, the accuracy performance deteriorates. Specifically, due to model deviation from actual pedestrian motion, the tracker keeps on making erroneous predictions, and when the corresponding target's measurement does not tally with this prediction, the algorithm terminates the track as evident by regular cardinality errors in the plots. Furthermore, the OSPA distances are relatively high as compared to the KITTI-17 plots.

4.2.3. Execution Results:

In a heterogeneous platform with an Intel Xeon processor and an AMD GPU W7100, we carried out the extensive profiling of the implemented SMC-LMB tracker algorithm. First, we extensively evaluated the execution performance of the C++ code to spot the potential performance bottlenecks. These were then subjected to OpenCL acceleration to come up with an implementation meeting the real-time constraints. We made use of the MOT videos (Table 1) to carry out this analysis.

For helping in spotting the potential bottlenecks, we split the main C++ algorithm into six parts, which are: (a) predict LMB; (b) predict δ -GLMB; (c) update LMB intermediate; (d) update δ -GLMB; (e) update LMB; (f) track-management and state-estimation.

Using the CPU clock-cycle metric as mentioned at the start of this section, we present our findings in Figure 7. These plots clearly show that the update LMB computation is the major bottleneck within the sequential SMC-LMB implementation. For KITTI-17, we get an average fps of 20 fps, while for the

worst-case (WC) frame, we get up to 5 fps. For a more tough video PETS09-S2L1, we rather get an average fps of 15 fps over all frames, while the WC frame corresponds to about 2 fps.

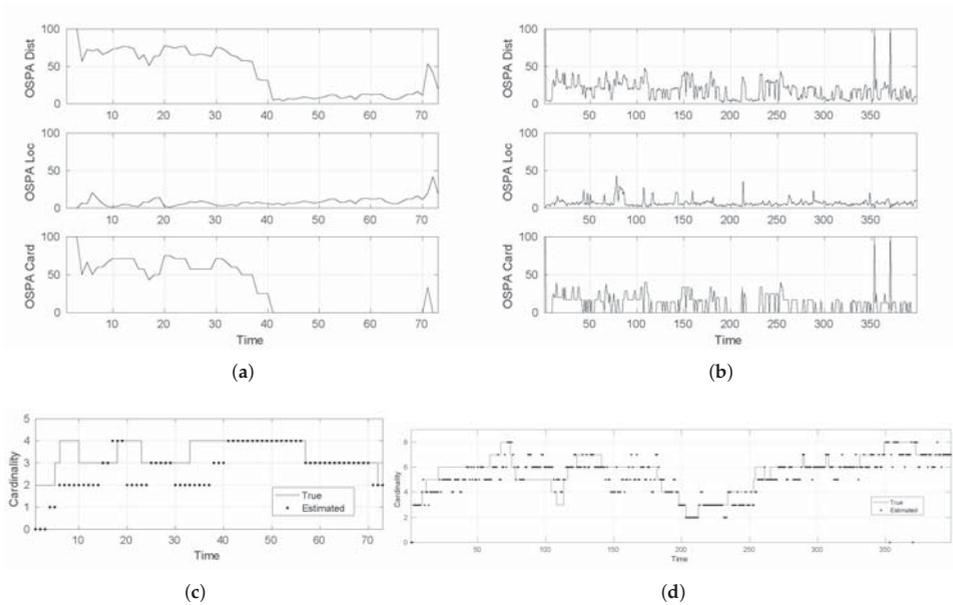


Figure 6. LMB tracker accuracy: MOT dataset videos. (a) OSPA distance KITT1-17; (b) OSPA distance PETS09-S2L1; (c) cardinality estimate KITT1-17; (d) cardinality estimate PETS09-S2L1.

As explained in Section 3, we target the update LMB computation using OpenCL compute kernels. Using this acceleration, we manage to gain substantial improvements in terms of execution performance, as shown in Figure 8, where for the PETS09-S2L1 video, we now get an average fps of 100 fps, while the WC frame amounts to 36 fps. Figure 9 further compares the OpenCL timings directly with the C++ ones to further show the substantial gains in performance for the two algorithmic functions that have been accelerated up till now.

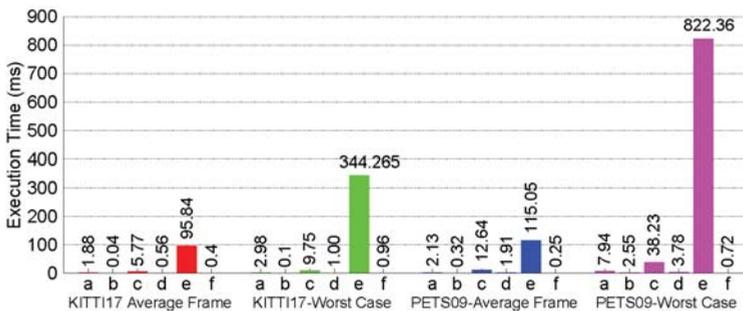


Figure 7. C++ LMB tracker execution performance.

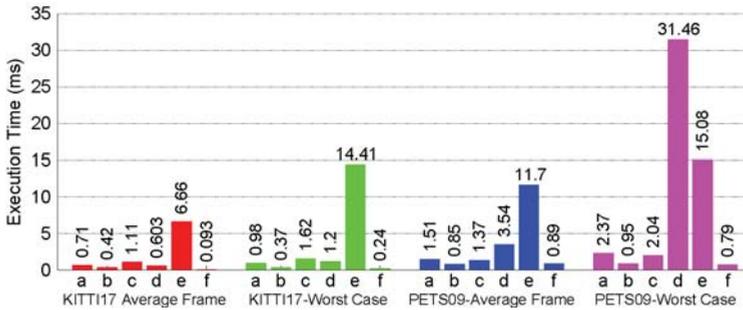


Figure 8. OpenCL accelerated LMB tracker execution performance.

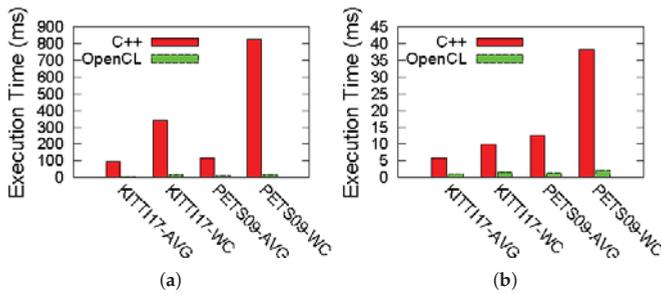


Figure 9. Comparison between C++ and OpenCL LMB implementations. (a) Update LMB computation, (b) Update LMB intermediate computation.

Finally, we also carried out the profiling of the LMB tracker algorithm under its different configurations. We studied the impact of changing its number of particles, birth components, survive components and update components. For this purpose, we used four different configurations (Table 2) to tackle a variant of the hard-scenario using $\lambda_c = 5$. These results are presented in Figure 10. As shown in these plots, the OpenCL implementation not only offers advantages in terms of sheer execution timings, but also provides a much more scalable implementation as compared to its pure C++ counterpart. The execution times rise nearly exponentially for both MOT videos in the case of the C++ version, while the rise is much less steep or rather linear for the case of the OpenCL-based implementation.

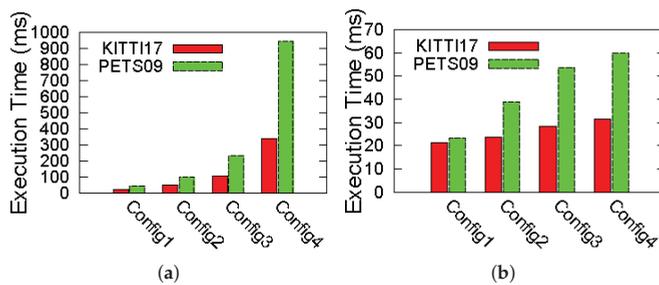


Figure 10. Scalability of C++ and OpenCL LMB implementations. (a) C++ implementation; (b) OpenCL implementation.

Table 2. LMB tracker configurations.

# Particles	# Births	# Survivals	# Updates
128	5	20	20
256	10	50	50
512	20	100	100
1024	50	200	200

At last, we have to note that there is still much room to improve the computation time while maintaining the tracking accuracy. In this paper, the prediction and update have been processed in two separate steps. This separation decreases the computation efficiency. In a recent paper, Vo et al. [5] proposed an efficient implementation by combining the prediction and update into a single step, which has a linear complexity in the number of hypothesized objects. It has been shown in [5] that the joint computation can speed up the execution time ranging from a dozen times to a thousand times for the linear Gaussian scenario. In our scenario, the joint computation can improve the performance on KITTI17 more than on PETS09 because most targets in KITTI17 move linearly. Figure 8 shows that the computation part of updating and predicting has taken 99% of the computation for KITTI17 and 95% for PETS09. It is therefore estimated that there will be around a 10- to 100-times computation speedup. The speedup effect on PETS09 will be less than on KITTI17 because targets in PETS09 move in a nonlinear way.

5. Conclusions

In this paper, we have developed two random finite set-based Bayesian filtering approaches, Gaussian mixture probability hypothesis density (GM-PHD) and labeled multi-Bernoulli (LMB) filters. The two approaches were designed in a highly modular way. After conducting their accuracy evaluations towards the multi-target tracking problem, we found that LMB filters were more appropriate to track the pedestrians. Then, we implemented in the LMB filter in C++ and carried out an extensive execution profiling. OpenCL programming was used to relieve the execution burden from the execution bottlenecks. The experimental results demonstrated a high computation improvement. In particular, the frame per second was improved from 15 fps to 100 fps on average, and the worst-case computation was also improved 18× from 2 fps to 36 fps.

Acknowledgments: This work was supported by the German Research Foundation (DFG) and the Technical University of Munich (TUM) in the framework of the Open Access Publishing Program.

Author Contributions: In this paper, Biao Hu and Kai Huang proposed the ideas. Uzair Sharif and Rajat Koner performed the experiments. Uzair Sharif also provided a masters thesis as the paper reference. Guang Chen and Biao Hu prepared the manuscript. Biao Hu and Kai Huang, Feihu Zhang, Walter Stechele and Alois Knoll provided the supervision and direction of the work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vo, B.N.; Ma, W.K. The Gaussian mixture probability hypothesis density filter. *IEEE Trans. Signal Process.* **2006**, *54*, 4091–4104.
2. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *14*, 28–44.
3. Leal-Taixé, L.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv* **2015**, arXiv:1504.01942.
4. Vo, B.N.; Vo, B.T.; Phung, D. Labeled random finite sets and the Bayes multi-target tracking filter. *IEEE Trans. Signal Process.* **2014**, *62*, 6554–6567.
5. Vo, B.N.; Vo, B.T.; Hoang, H. An Efficient Implementation of the Generalized Labeled Multi-Bernoulli Filter. *IEEE Trans. Signal Process.* **2016**, *65*, 1975–1987.
6. Bar-Shalom, Y. *Tracking and Data Association*; Academic Press Professional, Inc.: Cambridge, MA, USA, 1987.

7. Goodman, I.R.; Mahler, R.P.; Nguyen, H.T. *Mathematics of Data Fusion*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; Volume 37.
8. Mahler, R.P. "Statistics 101" for multisensor, multitarget data fusion. *IEEE Aerosp. Electron. Syst. Mag.* **2004**, *19*, 53–64.
9. Mahler, R. "Statistics 102" for multisource-multitarget detection and tracking. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 376–389.
10. Mahler, R.P. *Statistical Multisource-Multitarget Information Fusion*; Artech House, Inc.: Norwood, MA, USA, 2007.
11. Vo, B.T.; Vo, B.N.; Cantoni, A. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Trans. Signal Process.* **2009**, *57*, 409–423.
12. Vo, B.T.; Vo, B.N. Labeled random finite sets and multi-object conjugate priors. *IEEE Trans. Signal Process.* **2013**, *61*, 3460–3475.
13. Reuter, S.; Vo, B.T.; Vo, B.N.; Dietmayer, K. The labeled multi-Bernoulli filter. *IEEE Trans. Signal Process.* **2014**, *62*, 3246–3260.
14. Reuter, S.; Vo, B.T.; Vo, B.N.; Dietmayer, K. Multi-Object Tracking Using Labeled Multi-Bernoulli Random Finite Sets. In Proceedings of the 2014 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–8.
15. Dollár, P.; Appel, R.; Belongie, S.; Perona, P. Fast feature pyramids for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1532–1545.
16. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
17. Panta, K.; Clark, D.E.; Vo, B.N. Data association and track management for the Gaussian mixture probability hypothesis density filter. *IEEE Trans. Aerosp. Electron. Syst.* **2009**, *45*, doi:10.1109/TAES.2009.5259179.
18. Eppstein, D. Finding the k shortest paths. *SIAM J. Comput.* **1998**, *28*, 652–673.
19. Bellman, R. On a routing problem. *Q. Appl. Math.* **1958**, *16*, 87–90.
20. Murty, K.G. Letter to the editor—An algorithm for ranking all the assignments in order of increasing cost. *Oper. Res.* **1968**, *16*, 682–687.
21. Schuhmacher, D.; Vo, B.T.; Vo, B.N. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. Signal Process.* **2008**, *56*, 3447–3457.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

State Estimation Using Dependent Evidence Fusion: Application to Acoustic Resonance-Based Liquid Level Measurement

Xiaobin Xu ^{1,*}, Zhenghui Li ¹, Guo Li ¹ and Zhe Zhou ²

¹ School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China; 151060063@hdu.edu.cn (Z.L.); 162060112@hdu.edu.cn (G.L.)

² School of Engineering, Huzhou University, Huzhou 310027, China; zzhou@zjhu.edu.cn

* Correspondence: xuxiaobin1980@hdu.edu.cn; Tel.: +86-137-7759-8844

Academic Editors: Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang

Received: 15 February 2017; Accepted: 19 April 2017; Published: 21 April 2017

Abstract: Estimating the state of a dynamic system via noisy sensor measurement is a common problem in sensor methods and applications. Most state estimation methods assume that measurement noise and state perturbations can be modeled as random variables with known statistical properties. However in some practical applications, engineers can only get the range of noises, instead of the precise statistical distributions. Hence, in the framework of Dempster-Shafer (DS) evidence theory, a novel state estimation method by fusing dependent evidence generated from state equation, observation equation and the actual observations of the system states considering bounded noises is presented. It can be iteratively implemented to provide state estimation values calculated from fusion results at every time step. Finally, the proposed method is applied to a low-frequency acoustic resonance level gauge to obtain high-accuracy measurement results.

Keywords: DS evidence theory; state estimation; liquid level measurement; alarm monitoring

1. Introduction

Estimating the state of a dynamic system based on noisy sensor measurements is a common problem in sensor methods and applications [1,2]. Mainstream estimation methods all assume that both the system state noise and measurement noise can be modeled as random variables with known statistical properties. The Kalman filter, which supposes both of noises obey Gaussian distributions, is, by far, the most popular method [3]. The basic Kalman filter is only applicable to linear systems. In order to deal with nonlinear cases Bucy and Sunahara proposed the extended Kalman filter (EKF) [4,5]. The EKF uses the first order Taylor expansion technique to linearize state and observation equations, and then obtains state estimations by the Kalman filter. On the other hand, approximation to a state probability distribution of a nonlinear system is, to a great extent, easier and more feasible than a linear approximation to a nonlinear function [6]. Based on this idea, Gordon and Salmond proposed the particle filter (PF) [6]. The performance of the PF is commonly superior to the EKF because it can usually provide more precise information about state posterior probability distribution than does the EKF, especially when it takes a multimodal shape or noise distributions are non-Gaussian [6,7].

The precondition of the above methods is that the noise statistical properties must be known. However, in some practical applications, what engineers can obtain are not precise statistical distributions [8], but ranges of noises. Hence, a group of state estimation methods considering bounded noises, also known as the bounded-error methods, appeared [9–12]. Assuming that all variables belong to known compact sets, these methods build simple sets, such as ellipsoids or boxes, guaranteed to contain all state vectors consistent with given constraints. For linear systems, some scholars began to

study such state estimation methods in the 1960s [9–11]. For nonlinear systems, the corresponding studies are relatively rare. Khemani et al. and Jaulin proposed bounded-error state and parameter estimations for nonlinear systems [13,14]. Gning proposed a relatively simple and fast bounded-error method based on interval analysis and constraint propagation, which was successfully applied to dynamic vehicle localization [12], but when the noise bounds cannot be precisely determined, its robustness will unavoidably decline [7]. That is to say, if the bounds are too tight, then the data may become inconsistent with the system equations, and in this case, this method fails to provide a solution. On the contrary, if the bounds are overestimated, then the estimated state becomes very imprecise, and this method becomes overly pessimistic [7].

In order to deal with this problem, Nassreddine proposed an improved method by integrating interval analysis with DS evidence theory. Its key idea is to replace the set-based representation of uncertainty by a more general formalism, namely, mass functions in evidence theory [7]. It introduces possibility distributions to model bounded noises, and then uses mass functions, i.e., evidence composed of interval focal elements and their masses to approximate these distributions. Essentially, such mass functions can be regarded as “generalized boxes” composed of a collection of boxes with associated weights. These mass functions can be propagated in the system equations using interval arithmetic and constraint-satisfaction techniques to get the mass function of system state at each time step. Pignistic expectation of this mass function is calculated as the state estimation value. Therefore, this approach extends the pure interval approach, making it more robust and accurate.

Nassreddine’s research showed the powerful ability of DS evidence theory to deal with the uncertainty of dynamic systems. Hence, this paper further presents a new state estimation method, which uses not only evidential description of uncertainty, but also dependent evidence fusion. Here, the state equation and the observation equation of a dynamic system and the actual observations of system states are regarded as three information sources. The random set description of evidence and extension principle of random set is used to obtain state evidence and observation evidence from these three information sources and to propagate them in the system equations. There are correlation among these evidence, so the proposed combination rule of dependent evidence is used to fuse the propagated evidence and Pignistic expectation of fusion results is calculated as state estimation value at each time step. Compared with Nassreddine’s method, it is shown that the proposed approach generates more accurate estimation results by combining dependent evidence. An industrial liquid level detection apparatus was employed to show the better performance of the approach.

2. Foundations of Dempster-Shafer (DS) Evidence Theory

The DS theory is a mechanism formalized by Shafer for representing and reasoning with uncertain, imprecise, and incomplete information. It is initially based on Dempster’s original work on the modeling of uncertainty in terms of upper and lower probabilities induced by a multi-valued mapping rather than as a single probability value [15]. One of the specificities of this theory is that the objects of study are no more the universe, i.e., a set, defined as the frame of discernment hereinafter, but the power set of this universe. In this section we introduce some main concepts of this theory and some necessary notions that will be used in the proposed approach. A more detailed exposition and some background information can be found in [16].

2.1. Basic Concepts in DS Evidence Theory

Definition 1 (Frame of discernment). *A set is called a frame of discernment if it contains mutually exclusive and exhaustive possible hypotheses. This set is usually denoted as Θ . The power set of Θ is denoted as 2^Θ .*

Definition 2 (Mass function). *A function $m: 2^\Theta \rightarrow [0, 1]$ is called a mass function on Θ if it satisfies the following two conditions: (1) $m(\emptyset) = 0$; (2) $\sum_{A \in 2^\Theta} m(A) = 1$. This function is also named as a basic belief assignment (BBA). A subset A with a non-null mass is viewed as a focal element. Commonly, if an information*

source can provide a mass function on Θ , then this mass function is called a body of evidence, abbreviated to evidence (E).

Definition 3 (Dempster’s combination rule). If m_1, m_2 are two BBAs induced from two statistically independent information sources, then a combined BBA can be obtained by using Dempster’s combination rule:

$$m(A) = \begin{cases} \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B)m_2(C)}, & A \subseteq \Theta \text{ and } A \neq \emptyset \\ 0, & A = \emptyset \end{cases} \tag{1}$$

Note that the Dempster’s combination rule is meaningful only when $\sum_{B \cap C = \emptyset} m_1(B)m_2(C) < 1$, i.e., m_1 and m_2 are not totally conflicting. This rule can be used to synthesize uncertain, imprecise or incomplete information coming from different sources.

2.2. The Degree of Dependence and the Combination of Dependent Evidence

In DS evidence theory, Dempster’s combination rule is the most important tool for computing a new BBA from two BBAs based on two pieces of evidence. This rule requires that the two pieces of evidence must be independent, which is considered to be a very strong constraint and cannot always be met in practice. Wu, Yang and Liu [17] pointed that if there are two pieces of evidence which are partially derived from the same information source, then both of them are mutually dependent. This interpretation concentrates on the connotation of independence conception in evidence combination operation. In this case, Wu, Yang and Liu [17] proposed the energy of evidence concept, and then, deduced the degree of dependence and the dependency coefficient between the two from the energy of the intersection of the two. Based on these notions, the combination of dependent evidence can be realized.

Definition 4 (The energy of evidence E). The energy of evidence E, $En(E)$ is defined as:

$$En(E) = \sum_{\substack{i=1 \\ A_i \neq \emptyset}}^{n(E)} \frac{m(A_i)}{|A_i|} \tag{2}$$

where $|A_i|$ is the number of elements in the focal element A_i , $n(E)$ is the number of distinct focal elements in E. Obviously, $En(E)$ have some valuable characteristics: (1) if every $m(A_i) = 0$, namely, $m(\emptyset) = 1$, then $En(E) = 0$ and the evidence E represents no useful information; (2) if $|A_i| = 1$ and $m(\emptyset) = 0$, then $En(E) = 1$ and the E contains the maximum useful information; (3) $En(E) \in [0, 1]$.

Suppose that the BBAs of evidence E_1 and E_2 are m_1 and m_2 , respectively, and their focal element sequences are A_i and B_j . It is possible that some focal elements of E_1 and E_2 are induced by the same information source. In this case, E_1 and E_2 will be dependent, then the energy of the intersection of the two pieces of evidence can be described by:

$$En(E_1, E_2) = \sum_{\substack{ij=1 \\ D_{ij} \neq \emptyset}}^{|\{D_{ij}\}|} \frac{m(D_{ij})}{|D_{ij}|} \tag{3}$$

where D_{ij} denotes dependent focal element, $|\{D_{ij}\}|$ is the number of distinct D_{ij} and the BBA function m is derived from m_1 and m_2 .

The relationship of $En(E_1)$, $En(E_2)$ and $En(E_1, E_2)$ is illustrated in Figure 1; especially, $En(E_1, E_2) = 0$ implies the independence between E_1 and E_2 . The value of $En(E_1, E_2)$ measures the dependency of the two pieces of evidence.

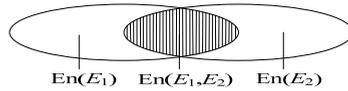


Figure 1. The relationships of two pieces of dependent evidence.

Definition 5 (The degree of dependence between two pieces of evidence). $En(E_1, E_2)$ is defined as the degree of dependence between E_1 and E_2 . Actually, the partial energies $En(E_1) - En(E_1, E_2)$ in E_1 and $En(E_2) - En(E_1, E_2)$ in E_2 are independent of each other. If energy $En(E_1, E_2)$ is partitioned into two parts, with each part attached to E_1 and E_2 , respectively, as follows:

$$D(E_1, E_2) = \frac{2En(E_1, E_2)}{En(E_1) + En(E_2)} \tag{4}$$

then two corresponding independent pieces of evidence can be generated from E_1 and E_2 .

For E_1 , its final independent energy can be calculated as:

$$\begin{aligned} En_f(E_1) &= En(E_1) - En(E_1, E_2) + En(E_1, E_2) \frac{En(E_1)}{En(E_1) + En(E_2)} \\ &= En(E_1) - En(E_1, E_2) \frac{En(E_2)}{En(E_1) + En(E_2)} \\ &= En(E_1) \left(1 - \frac{En(E_1, E_2)}{En(E_1) + En(E_2)} * \frac{En(E_2)}{En(E_1)} \right) \\ &= En(E_1) \left(1 - \frac{1}{2} D(E_1, E_2) * \frac{En(E_2)}{En(E_1)} \right) \end{aligned} \tag{5}$$

Similarly:

$$En_f(E_2) = En(E_2) \left(1 - \frac{1}{2} D(E_1, E_2) * \frac{En(E_1)}{En(E_2)} \right) \tag{6}$$

Definition 6 (The dependency coefficient between two pieces of evidence). The dependency coefficient of E_1 to E_2 is defined as:

$$R_{12} = \frac{1}{2} D(E_1, E_2) \frac{En(E_2)}{En(E_1)} \tag{7}$$

and the dependency coefficient of E_2 to E_1 is defined as:

$$R_{21} = \frac{1}{2} D(E_1, E_2) \frac{En(E_1)}{En(E_2)} \tag{8}$$

E_1 and E_2 can be modified by R_{12} and R_{21} , respectively, to obtain their corresponding independent E_1' and E_2' , their BBA functions are given by:

$$m_1'(A) = \begin{cases} (1 - R_{12})m_1(A), & \forall A \subseteq \Theta, A \neq \Theta \\ 1 - \sum_{A \subset \Theta} m_1'(A) & A = \Theta \end{cases} \tag{9}$$

$$m_2'(B) = \begin{cases} (1 - R_{21})m_2(B), & \forall B \subseteq \Theta, B \neq \Theta \\ 1 - \sum_{B \subset \Theta} m_2'(B) & B = \Theta \end{cases} \tag{10}$$

Consequently, the requirement of Dempster's rule is met and the combination of E_1' and E_2' can be implemented according to Dempster's rule in (1). Finally, the combination of E_1 and E_2 is indirectly realized by the combination of E_1' and E_2' . Actually, reference [17] gives the decorrelation method to correct E_1 and E_2 by dependency coefficients such that the corrected E_1' and E_2' can be deemed as the independent evidence and combined using Dempster's rule.

2.3. The Random Set Description of Evidence

2.3.1. Random Set and Random Relation

Definition 7 (Random set [18,19]). A finite support random set on Θ is a pair (\mathcal{F}, m) where \mathcal{F} is a finite family of distinct non-empty subsets of Θ and m is a mapping $\mathcal{F} \rightarrow [0, 1]$ and such that $\sum_{A \in \mathcal{F}} m(A) = 1$.

\mathcal{F} is called the support of the random set and m is called a basic belief assignment. Such a random set (\mathcal{F}, m) is equivalent to a mass function in the sense of Shafer.

Definition 8 (Random relation [18,19]). Let $\Theta = \Theta_1 \times \Theta_2 \times \dots \times \Theta_n$ be a multi-dimensional space, where “ \times ” indicates Cartesian product. A finite support random relation is a random set (\mathcal{F}, m) on Θ .

The projections of a random relation on Cartesian product $\Theta_1 \times \Theta_2 \times \dots \times \Theta_n$ are defined by Shafer to be the marginal random set (\mathcal{F}_k, m_k) ($k = 1, 2, \dots, n$):

$$\forall C_k \subseteq \Theta_k, m_k(C_k) = \sum \left\{ m(A) \mid C_k = \text{Proj}_{\Theta_k}(A) \right\} \tag{11}$$

$$\text{Proj}_{\Theta_k}(A) = \{ u_k \in \Theta_k \mid \exists u = (u_1, \dots, u_k, \dots, u_n) \in A \} \tag{12}$$

For $\forall A \in \mathcal{F}, A = C_1 \times C_2 \times \dots \times C_n$, if $m(A) = m_1(C_1) \times m_2(C_2) \times \dots \times m_n(C_n)$, then (\mathcal{F}, m) is called decomposable Cartesian product random relation, and marginal random sets $(\mathcal{F}_1, m_1), (\mathcal{F}_2, m_2), \dots, (\mathcal{F}_n, m_n)$ are mutually independent.

2.3.2. Extension Principles

Let $\zeta = (\zeta_1, \zeta_2, \dots, \zeta_n)$ be a variable on $\Theta = \Theta_1 \times \Theta_2 \times \dots \times \Theta_n, \zeta = f(\xi), \xi \in \Phi, f : \Theta \rightarrow \Phi$ is the function of ζ . The random set (\mathcal{R}, ρ) of ζ , which is the image of random relation (\mathcal{F}, m) of ζ through f , is given by extension principles [20–22]:

$$\mathcal{R} = \{ R_i = f(A_i) \mid A_i \in \mathcal{F} \} \tag{13}$$

$$\rho(R_j) = \sum \{ m(A_i) \mid R_j = f(A_i) \} \tag{14}$$

where:

$$f(A_i) = \{ f(u) \mid u \in A_i \}, i = 1, 2, \dots, M \tag{15}$$

M is the number of element of \mathcal{F} . The summation in Equation (14) accounts for the fact that more than one focal element A_i may yield the same image R_j through f .

The key of constructing (\mathcal{R}, ρ) is to calculate the image of A_i through f . If ζ is a continuous variable on Θ , then $\Theta = \mathbb{R}^n, \mathcal{F}$ becomes a finite family of distinct non-empty sub-intervals on Θ . In this case, the process of constructing (\mathcal{R}, ρ) is given as follows:

For each ζ_k in ζ , let its marginal random set be (\mathcal{F}_k, m_k) and the focal element of (\mathcal{F}_k, m_k) be a interval $[a_k^-, a_k^+]$, then the focal element of (\mathcal{F}, m) can be given as:

$$A = [a_1^-, a_1^+] \times \dots \times [a_n^-, a_n^+] \tag{16}$$

The image of A can be calculated by using the methods of Interval Analysis [19–21]; if A is a convex set, then A has 2^n vertices, denoted as v_j ($j = 1, 2, \dots, 2^n$). If function f has certain properties, the Vertex Method can help reduce the calculation time considerably [22]:

Proposition 1. $\forall A \in \mathcal{F}$, if $\zeta = f(\xi)$ is continuous in A and also no extreme point exists in this region (including its boundaries), then the value of interval function can be obtained by:

$$f(A) = R = \left[\min_j \{ f(v_j) : j = 1, 2, \dots, 2^n \}, \max_j \{ f(v_j) : j = 1, 2, \dots, 2^n \} \right] \tag{17}$$

Thus, function f has to be evaluated 2^n times for each focal element A . This computational burden can be further reduced if the hypotheses of the following Proposition 2 hold [21].

Proposition 2. If f is continuous, if its partial derivatives are also continuous and if f is a strictly monotonic function with respect to each $\xi_k, k = 1, 2, \dots, n$, then:

$$\exists v_{\min}, f(v_{\min}) = \min_j \{f(v_j) : j = 1, 2, \dots, 2^n\} \tag{18}$$

$$\exists v_{\max}, f(v_{\max}) = \max_j \{f(v_j) : j = 1, 2, \dots, 2^n\} \tag{19}$$

There is a case in point. Let $\xi = (\xi_1, \xi_2, \xi_3)$, $A = [a_1^-, a_1^+] \times [a_2^-, a_2^+] \times [a_3^-, a_3^+]$. Assume f and its partial derivatives are all continuous. If f is increasing with respect to ξ_1 and ξ_2 , decreasing with respect to ξ_3 respectively, then f has to be calculated only twice for each focal element A , namely, $f(A) = [f(v_{\min}), f(v_{\max})]$, $v_{\min} = [a_1^-, a_2^-, a_3^+]$, $v_{\max} = [a_1^+, a_2^+, a_3^-]$. Totally, $2M$ evaluations of f are needed to obtain complete (\mathcal{R}, ρ) .

Furthermore, the expectation of (\mathcal{R}, ρ) is given by [23]:

$$\mathbb{E}(\rho) = \sum_{j=1}^N \rho(R_j) \cdot \left(\frac{r_j^+ + r_j^-}{2} \right) \tag{20}$$

where $R_j = [r_j^-, r_j^+]$, $j = 1, 2, \dots, N$, N is the number of focal element R_j .

3. State Estimation Based on Dependent Evidence Fusion

3.1. Dynamic System Model under Bounded Noises

The dynamic systems mode constructed by the state and observation equations is as follows:

$$\begin{cases} x_{k+1} = f(x_k, v_k) \\ z_{k+1} = g(x_{k+1}, w_{k+1}) \end{cases} \quad k = 1, 2, 3, \dots \tag{21}$$

where the relationship between state x_{k+1} at time $k + 1$ and state x_k at time k is described as function f . The relationship between observation z_{k+1} at time $k + 1$ and state x_{k+1} at time $k + 1$ is described as function g . v_k and w_k are bounded additive state noise variable and observation noise variable respectively, which are independent of each other. These two noises can be approximated to triangle possibility distributions [7], noted as π_v and π_w , respectively, (the noise distributions are identical at each time step), as shown in Figure 2.

$$\pi_v(v) = \begin{cases} \frac{v-v_a}{v_c-v_a} & \text{if } v_a \leq v \leq v_c \\ \frac{v_b-v}{v_b-v_c} & \text{if } v_c \leq v \leq v_b \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

where $[v_a, v_b]$ is the support interval of the state noise, v_c is the mode of state noise, similarly:

$$\pi_w(w) = \begin{cases} \frac{w-w_a}{w_c-w_a} & \text{if } w_a \leq w \leq w_c \\ \frac{w_b-w}{w_b-w_c} & \text{if } w_c \leq w \leq w_b \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

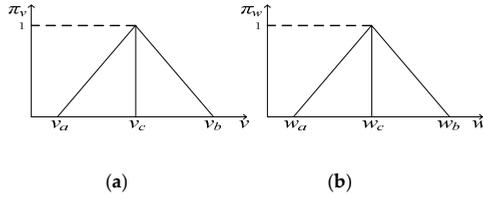


Figure 2. (a) Triangle possibility distributions of state noises, (b) Triangle possibility distributions of observation noises.

3.2. Recursive Algorithm of State Estimation Based on Extension Principles and Dependent Evidence Fusion

Figure 3 shows the flow of the proposed recursive algorithm. The following steps will be introduced in detail.

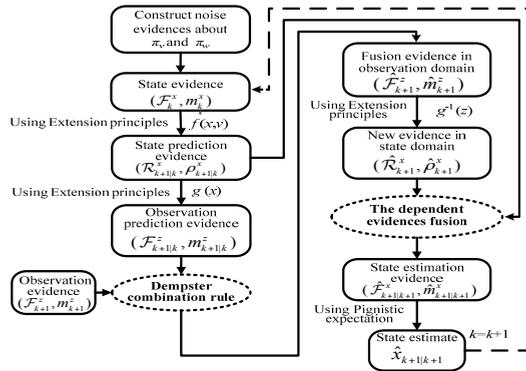


Figure 3. Flowchart of state estimation iterative algorithm.

Step 1: Construct noise evidence to approximate π_v and π_w . Initially, we construct evidence (\mathcal{F}_k^v, m_k^v) to approximate the possibility distribution π_v of state noise variable v_k . For any $\alpha \in (0, 1]$, α cut set of π_v is [9]:

$$[\pi_{\alpha}^{v-}, \pi_{\alpha}^{v+}] = \{v | \pi_v(v) \geq \alpha\} \tag{24}$$

If there exist $\alpha_0, \alpha_1, \dots, \alpha_{p-1}$ which satisfy $0 = \alpha_0 < \alpha_1 < \dots < \alpha_{p-1} < 1$, then their corresponding α -cut sets will satisfy $[\pi_{\alpha_0}^{v-}, \pi_{\alpha_0}^{v+}] \subset [\pi_{\alpha_1}^{v-}, \pi_{\alpha_1}^{v+}] \subset \dots \subset [\pi_{\alpha_{p-1}}^{v-}, \pi_{\alpha_{p-1}}^{v+}]$ where p is a positive integer. Take these p α -cut sets as focal elements with nested closed interval forms, then their corresponding BBAs are:

$$m([\pi_{\alpha_i}^{v-}, \pi_{\alpha_i}^{v+}]) = \begin{cases} \alpha_{i+1} & \text{if } i = 0 \\ \alpha_{i+1} - \alpha_i & \text{if } i = 1, 2, \dots, p - 2 \\ 1 - \alpha_i & \text{if } i = p - 1 \end{cases} \tag{25}$$

Figure 4 gives an example that when $p = 3$, and m can be constructed by uniformly cutting α three times. Distinctly, m corresponds to a possibility distribution π that approximates π_v . Certainly, a better approximation of the continuous possibility distribution can be obtained by increasing the number p of cut sets, at the expense of higher computational complexity.

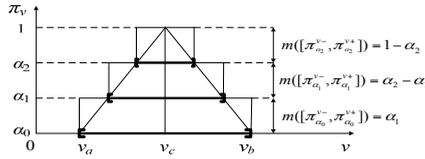


Figure 4. The possibility distribution of state noise and its evidence construction.

It is worth noticing that m is constructed on the condition that all values outside the support interval $[v_a, v_b]$ are completely impossible. However, in practice, the bounds v_a and v_b are commonly given based on available measurement knowledge or real data, so they may not be precise and the values outside $[v_a, v_b]$ may appear. To account for the imprecision of the support interval [7], constructs (\mathcal{F}_k^v, m_k^v) by discounting m with a small discount rate ε_v , in which, m_k^v is defined as [9]:

$$m_k^v(A) = \begin{cases} (1 - \varepsilon_v)m(A) & \text{if } A \in \Xi \\ (1 - \varepsilon_v)m(\Theta) + \varepsilon_v & \text{if } A = \Theta \end{cases} \quad (26)$$

where $\Xi = \{[\pi_{\alpha_i}^-, \pi_{\alpha_i}^+] | i = 1, 2, \dots, p - 1\}$, $\Theta = \mathbb{R}$, accordingly, $\mathcal{F}_k^v = \Xi \cup \Theta$. In the course of implementing the proposed algorithm, Θ can be replaced by the closed interval $[v_a', v_b']$, here $v_a' \gg v_a$ and $v_b' \gg v_b$ such that the following interval operations can be done easily.

In the same way, we can construct evidence (\mathcal{F}_k^w, m_k^w) using the possibility distribution π_w of observation noise variable w_k .

Step 2: Obtain state prediction evidence. $E_{k+1|k}^x = (R_{k+1|k}^x, \rho_{k+1|k}^x)$ at time $k + 1$ from state equation. Suppose the estimation result at time k is $\hat{x}_{k|k}$. When $k = 1$, $\hat{x}_{1|1}$ is initialized as real observation z_1 . Considering the influence of noise to the state, we construct the state evidence (\mathcal{F}_k^x, m_k^x) of $\hat{x}_{k|k}$ by adding noise to $\hat{x}_{k|k}$:

$$\mathcal{F}_k^x = \{[\pi_{\alpha_0}^v - \hat{x}_{k|k}, \pi_{\alpha_0}^v + \hat{x}_{k|k}], [\pi_{\alpha_1}^v - \hat{x}_{k|k}, \pi_{\alpha_1}^v + \hat{x}_{k|k}], \dots, [\pi_{\alpha_{p-1}}^v - \hat{x}_{k|k}, \pi_{\alpha_{p-1}}^v + \hat{x}_{k|k}], \Theta\}; \quad m_k^x = m_k^v$$

Thus, taking (\mathcal{F}_k^v, m_k^v) and (\mathcal{F}_k^x, m_k^x) as the inputs of state equation $x_{k+1} = f(x_k, v_k)$, we can get the state prediction evidence $E_{k+1|k}^x = (R_{k+1|k}^x, \rho_{k+1|k}^x)$ by mapping from the inputs to the outputs based on the extension principles in Equations (13) and (14).

Step 3: Obtain observation prediction evidence. $E_{k+1|k}^z = (\mathcal{F}_{k+1|k}^z, m_{k+1|k}^z)$ at time $k + 1$ from observation equation.

Taking the state prediction evidence $(R_{k+1|k}^x, \rho_{k+1|k}^x)$ in Step 2 as the input of the observation equation $g(x_{k+1})$, we can get $E_{k+1|k}^z = (\mathcal{F}_{k+1|k}^z, m_{k+1|k}^z)$ based on the extension principles in (13) and (14).

Step 4: Obtain fusion evidence. $\hat{E}_{k+1}^z = (\hat{\mathcal{F}}_{k+1}^z, \hat{m}_{k+1}^z)$ at time $k + 1$ in observation domain.

Firstly, in Step 1, we get evidence (\mathcal{F}_k^w, m_k^w) using the possibility distribution π_w of w_k :

$$\mathcal{F}_k^w = \{[\pi_{\alpha_0}^w - \pi_{\alpha_0}^w, \pi_{\alpha_0}^w + \pi_{\alpha_0}^w], [\pi_{\alpha_1}^w - \pi_{\alpha_1}^w, \pi_{\alpha_1}^w + \pi_{\alpha_1}^w], \dots, [\pi_{\alpha_{p-1}}^w - \pi_{\alpha_{p-1}}^w, \pi_{\alpha_{p-1}}^w + \pi_{\alpha_{p-1}}^w], \Theta\}; \quad m_k^w = m_k^v$$

After getting observation z_{k+1} at time $k + 1$, considering the influence of noise to the observation, we construct the evidence $(\mathcal{F}_{k+1}^z, m_{k+1}^z)$ of z_{k+1} through adding noise to z_{k+1} :

$$\mathcal{F}_{k+1}^z = \{[\pi_{\alpha_0}^z - z_{k+1}, \pi_{\alpha_0}^z + z_{k+1}], [\pi_{\alpha_1}^z - z_{k+1}, \pi_{\alpha_1}^z + z_{k+1}], \dots, [\pi_{\alpha_{p-1}}^z - z_{k+1}, \pi_{\alpha_{p-1}}^z + z_{k+1}], \Theta\}; \quad m_k^z = m_k^w$$

Secondly, using Dempster's combination rule, we fuse $(\mathcal{F}_{k+1}^z, m_{k+1}^z)$ and $(\mathcal{F}_{k+1|k}^z, m_{k+1|k}^z)$ to get the fusion evidence $\hat{E}_{k+1}^z = (\hat{\mathcal{F}}_{k+1}^z, \hat{m}_{k+1}^z)$ in observation domain at time $k + 1$. As for the relationship between $(\mathcal{F}_{k+1|k}^z, m_{k+1|k}^z)$ and $(\mathcal{F}_{k+1}^z, m_{k+1}^z)$. The former is obtained by propagating $\hat{x}_{k|k}$ from state

equation $f(x_k, v_k)$ to observation equation $g(x_{k+1})$; the latter is constructed by adding noise $\pi_w(w)$ to z_{k+1} . It can be seen that the former completely comes from the state information $\hat{x}_{k|k}$ at past time step k which does not use the observation noise $w_{k+1}(\pi_w(w))$, but uses the state noise $v_k(\pi_v(v))$. Because $w_{k+1}(\pi_w(w))$ and $v_k(\pi_v(v))$ are independent of each other, so it is believed that the former and the latter are also independent of each other. Hence both of them can be directly fused using Dempster's combination rule.

Step 5: Get new evidence. $\hat{E}_{k+1}^x = (\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x)$ at time $k+1$ in state domain.

Taking $\hat{\mathcal{F}}_{k+1}^z, \hat{m}_{k+1}^z$ attained in the Step 4 as the input of inverse function $g^{-1}(z_{k+1})$, we can get $(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x)$ by using the extension principles in Equations (13) and (14).

Step 6: Get state estimation evidence. $(\hat{\mathcal{F}}_{k+1|k+1}^x, \hat{m}_{k+1|k+1}^x)$ and state estimate $\hat{x}_{k+1|k+1}$ at time $k+1$.

Using Dempster's combination rule, we can fuse $(\mathcal{R}_{k+1}^x, \rho_{k+1}^x)$ attained in Step 5 and $(\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x)$ attained in Step 2. That is to say, we utilize the former to revise the latter to get state estimation evidence $(\hat{\mathcal{F}}_{k+1|k+1}^x, \hat{m}_{k+1|k+1}^x)$. $(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x)$ is obtained by inverse mapping of fusion evidence $(\hat{\mathcal{F}}_{k+1}^z, \hat{m}_{k+1}^z)$ in observation domain. $(\hat{\mathcal{F}}_{k+1}^z, \hat{m}_{k+1}^z)$ is obtained by the fusion of observation evidence $(\mathcal{F}_{k+1}^z, m_{k+1}^z)$ and observation prediction evidence $(\mathcal{F}_{k+1|k}^z, m_{k+1|k}^z)$. In Step 3, it is noted that $(\mathcal{F}_{k+1|k}^z, m_{k+1|k}^z)$ is related to $(\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x)$, so $(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x)$ and $(\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x)$ are certainly mutually dependent. Therefore, the combination of dependent evidence must be used for fusing both of them. For the focal elements of $(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x)$ and $(\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x)$ are the closed intervals on real numbers, here we extend the combination of dependent evidence in the discrete frame of discernment introduced in Section 2.2 to that in the continuous frame of discernment (see the corresponding proposition and example in Appendix A). $(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x)$ and $(\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x)$ can be fused using the extended combination of dependent evidence to get state estimation evidence $(\hat{\mathcal{F}}_{k+1|k+1}^x, \hat{m}_{k+1|k+1}^x)$ at time $k+1$.

Finally, Pignistic expectation of $(\hat{\mathcal{F}}_{k+1|k+1}^x, \hat{m}_{k+1|k+1}^x)$ is calculated as state estimation value $\hat{x}_{k+1|k+1}$ by Equation (20). Using state estimation at time $k+1$ to do next iteration, we can estimate state at every time step.

In conclusion, as shown in Figure 3, the whole recursive algorithm is actualized under the framework of DS evidence theory. The corresponding evidence in state and observation domains are not only propagated and transformed by the extension principle, but also fused by the Dempster's combination rule and the proposed combination rule for dependent evidence. Especially, fusion procedure can make that the masses focus to those interval focal elements that contain the system state, so as to get the accurate estimation results, which is the main difference from Nassreddine's method under the framework of the interval analysis. In next section, our approach will be applied to liquid level estimation using an industrial level apparatus to show its better performance than possible with Nassreddine's method.

4. Application to Liquid Level Measurement

Level measurement methods based on sound reflection phenomena have been successfully applied in some areas of process industry (chemical, waste water treatment, petroleum, etc.) because the level is the main monitored process variable used in industrial alarm systems. Ultrasonic measurement methods, with good directivity, convenient operation and so on, have become some of the most commonly used techniques [24]. Their measuring principle is to emit an ultrasound toward a liquid surface and receive the echos, then to calculate the distance from the surface to the acoustic receiver device by multiplying the sound velocity by the round-trip time [25]. However, this method is susceptible to the quality of the instrument itself and environmental noise, which will deteriorate the measurement accuracy. Besides, if the ultrasound encounters foams, residues, deposits, etc., in the measurement process, it is also prone to parasitic reflection, thereby the ultrasound propagation path is changed, which seriously affects the measurement accuracy [26].

On the contrary, low-frequency sound waves have longer wavelength and it is easy to generate the diffraction phenomenon which can effectively overcome the problem of parasitic reflection due to foams, residues, deposits, etc. When a speaker emits sound waves with a uniform change from a frequency f_L to a higher frequency f_H toward the surface and a microphone receives the corresponding echoes, the generated standing wave signals extracted in the oscilloscope can be used to calculate the height of the liquid level. Kumperščak and Završnik [25,26] used this idea to measure liquid levels. However, they both directly used observations to calculate the liquid level. In practice, if the measurements obtained by using a speaker and a microphone are not precise enough and if the effect of environmental noise is inevitable, then the deviation of the final measurement results will be unacceptable, which is the most common shortcoming in the present level measurement methods.

In our earlier work [27], we have used the Evidential Reasoning(ER) rule to deal with liquid level estimation with bounded noises, but the ER-based method only provides an initial idea for state estimation under the framework of DS evidence theory and only gives precise estimated results when the level length is less than 1.6 m. In order to improve the evidence fusion-based state estimate method, this paper introduces a new information source, Dempster combination rule and evidence dependence conceptions. We construct the state equation and observation equation based on the principle of level measurement using acoustic standing waves, and then use the proposed algorithm to estimate the frequencies of the standing waves, which can be translated into the liquid level height (0 m–10 m). Compared with the direct measurement method and Nassreddine's method, the estimation results verify that our algorithm has obvious advantages and improves the level estimation accuracy.

4.1. Acoustic Standing Wave Level Gauge

The structure of an acoustic standing wave level gauge is shown in Figure 5, and mainly consists of a waveguide (a tube), a speaker, a microphone, a thermometer and a controller. When sound waves in the frequency range $[f_L, f_H]$ generated by a signal generator (audio card and speaker) vertically propagate to a surface and echoes appear, superposition of both waves will generate standing waves. Here, y_1 denotes the sound wave generated by speaker and y_2 denotes the echo reflected by the surface:

$$y_1 = A \cos 2\pi \left(Pt - \frac{L}{\lambda} \right) \quad (27)$$

$$y_2 = A \cos 2\pi \left(Pt + \frac{L}{\lambda} \right) \quad (28)$$

The synthesis wave of y_1 and y_2 can be expressed as:

$$y = 2A \cos \left(\pi \frac{2L}{\lambda} \right) \cos(2\pi Pt) \quad (29)$$

where A is the amplitude of sound wave, P is the frequency of sound wave and L is the distance from the top of the tube to the surface of liquid as shown in Figure 5. From Equation (29), we know that when L and λ have the following relation, the amplitude of synthesis wave reaches the maximum:

$$L = n_k \frac{\lambda_k}{2} \quad k = 1, 2, 3, \dots \quad (30)$$

In this case, this synthesis wave is defined as the standing wave and its wavelength is:

$$\lambda_k = \frac{c}{f_k} = \frac{331.4 + 0.6T}{f_k} \quad (31)$$

where λ_k is the wavelength of k th standing wave, f_k is the frequency of k th standing wave (k th resonance frequency) in $[f_L, f_H]$. c is sound velocity, and T is temperature.

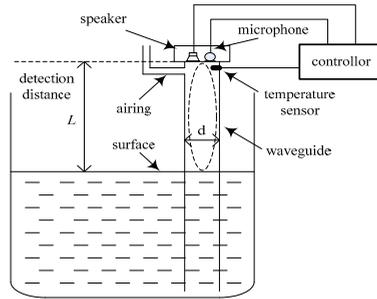


Figure 5. Structure of a level gauge.

Substituting Equation (30) into Equation (31), we obtain:

$$L = \frac{n_k(331.4 + 0.6T)}{2f_k} \quad (32)$$

where, n_k is given as [28]:

$$n_k = \frac{f_k}{(f_{k+1} - f_k)} \quad (33)$$

and:

$$n_{k+1} = n_k + 1 \quad (34)$$

Theoretically, in Equation (33), $f_{k+1} - f_k = f_F$, f_F is the fundamental resonance frequency and $f_k = n_k f_F$, $n_k \in \mathbb{N}^+$ (the set of all positive integers) [26,28]. For example, if $L = 9.6$ m, $T = 23.9$ °C, and $n = 1$, then the fundamental resonance frequency can be calculated by (32):

$$f_F = \frac{n(331.4 + 0.6T)}{2L} = 18\text{Hz} \quad (35)$$

If the frequency range $[f_L, f_H]$ is [1000 Hz, 2500 Hz], then there are 82 resonance frequencies in this range, $k = 1, 2, \dots, 82$, and $n_k = 56, 57, \dots, 137$. Consequentially, $f_1 = 56 \times 18$ Hz, $f_2 = 57 \times 18$ Hz, \dots , $f_{82} = 137 \times 18$ Hz.

4.2. System Model

Firstly, we consider the resonance frequency as the estimated state and construct the corresponding state equation. If we can continuously collect the resonance frequency f_{k+1} , then we have the following equations:

$$L = \frac{n_{k+1}(331.4 + 0.6T)}{2f_{k+1}} \quad (36)$$

From Equations (32) and (36), obviously, we can get:

$$\frac{n_{k+1}(331.4 + 0.6T)}{2f_{k+1}} = \frac{n_k(331.4 + 0.6T)}{2f_k} \quad (37)$$

$$f_{k+1} = \frac{n_k + 1}{n_k} f_k \quad (38)$$

Consequentially, we can establish the recursive linear state equation and observation equation, respectively:

$$x_{k+1} = \frac{n_k + 1}{n_k} x_k + v_k \quad (39)$$

$$z_{k+1} = x_{k+1} + w_{k+1} \quad (40)$$

where $x_k = f_k$, z_k is the observation of f_k , w_k and v_k are independent noise sequences coming from speaker and microphone, respectively, satisfying the conditions:

$$v_k = [v_a, v_b] \quad (41)$$

$$w_k = [w_a, w_b] \quad (42)$$

The intervals $[v_a, v_b]$ and $[w_a, w_b]$ denote the boundaries of the state noise and observation noise, respectively. The state noise v_k and observation noise w_k can be expressed by possibility distributions π_v and π_w with the support intervals $[v_a, v_b]$ and $[w_a, w_b]$, respectively.

It should be noted that, in theory, n_k in (39) should be taken as a positive integer. However, in practice, it can be only calculated by observations z_k and z_{k+1} according to Equation (33). Because of observation imprecision, the calculated n_k is commonly not a positive integer, so it should be approximated as:

$$n_k = \left\| \frac{z_k}{(z_{k+1} - z_k)} \right\| \quad (43)$$

where “ $\|\bullet\|$ ” denotes the operator that “round numbers to the nearest integer”.

4.3. Liquid Level Estimation Tests

In order to construct the level gauge in Figure 5, we use a low-precision microphone and speaker to emit and receive cosine sound waves, respectively, an electronic thermometer to collect temperature and a PVC tube with a diameter of $d = 75$ mm to transmit sounds. The estimated level L is the distance from the surface to the speaker platform. The controller transmits sine or cosine waves to drive the speaker to emit the signals vertically to the liquid surface. We use the software AUDIOSCSI (Brothers Studio, Shenzhen, China) which is based on an audio controller (82801HBM-ICH8M with sampling rate 44,100 Hz, Intel Corporation, Santa Clara, CA, USA) to generate sound waves. The frequencies of wave change with uniform speed from $f_L = 1000$ Hz to $f_H = 2400$ Hz in 5 s. Thus, the microphone receives the synthesis waves and sends them to the controller as shown in Figure 6 ($L = 4.6$ m). It can be seen that there are the frequencies of 39 adjacent standing waves collected by microphone in [1000 Hz, 2500 Hz]. Figure 7 shows the resonance frequency f_k ($k = 1, 2, \dots, 39$) extracted from the spectrum of the synthesis waves by the fast detecting algorithm in [28]. In this experiment, set liquid level distance $L = 4.6$ m, the ambient temperature is 26.5 °C and sound velocity is 347.3 m/s. The state equation and observation equation of resonance frequency are shown in Equations (39) and (40).

For the state noise v_k , we use a high-precision oscillograph (TPS2024, Tektronix, Shanghai, China) to receive the cosine sound waves emitted by the audio controller and speaker in range [1000 Hz, 2500 Hz] and calculate errors about 100 frequency points uniformly selected from 1000 Hz to 2500 Hz. As in [7], the bounds of v_k are taken to be plus or minus three times the standard deviation of errors. So the possibility distributions π_v of v_k can be constructed as in Figure 8. Where the expectation of π_v is 0, standard deviation σ_v is 0.1, then the support intervals $[v_a, v_b] = [-0.3, 0.3]$, mode $v_c = 0$. Set $\alpha_0 = 0$, $\alpha_1 = 1/3$, $\alpha_2 = 2/3$, we can get three nested closed intervals and their BBAs to approximate π_v as in Figure 8.

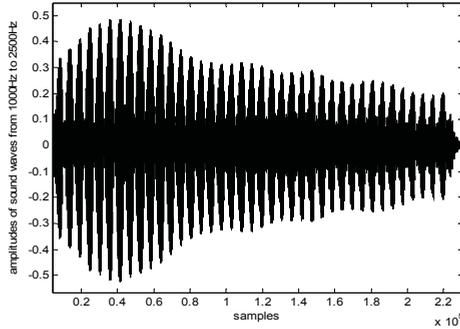


Figure 6. Waveform graph ($L = 4.6$ m).

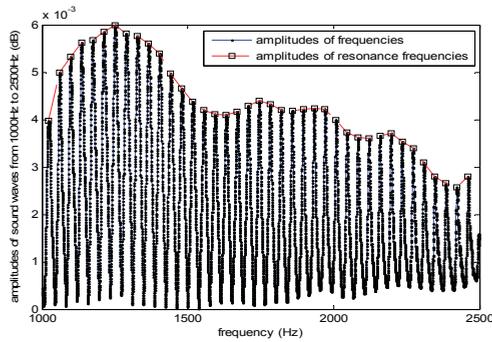


Figure 7. Resonance frequencies and amplitudes ($L = 4.6$ m).

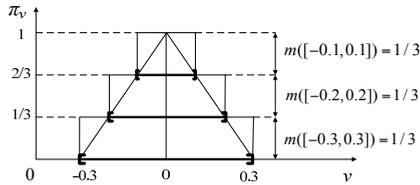


Figure 8. Probability distribution π_v of state noise v .

Furthermore, in Step 1 of the proposed algorithm in Section 3.2, discounting m at rate $\varepsilon_v = 0.05$ and approximating as the closed interval $[v_c - 100\sigma_v, v_c + 100\sigma_v] = [-10, 10]$, we can construct the evidence (\mathcal{F}_k^v, m_k^v) of v_k according to Equation (26) as shown in Table 1.

Table 1. Evidence of state noise.

\mathcal{F}_k^v	$[-0.1, 0.1]$	$[-0.2, 0.2]$	$[-0.3, 0.3]$	$[-10, 10]$
m_k^v	0.3167	0.3167	0.3167	0.05

The observation noise w_k is mainly related to the microphone and the fast detecting algorithm. Firstly, we extract observation values of resonance frequencies in range [1000 Hz, 2500 Hz] about 30 level values uniformly selected from $L = 1.3$ m to $L = 10.6$ m by the fast detecting algorithm. Secondly, we calculate the errors between the theoretically correct values (true values) and observation values.

In the same way, the possibility distributions π_w of w_k and the corresponding closed intervals and their BBAs can be constructed as in Figure 9, where, $\sigma_w = 1.23$, $w_c = -6.9$, so $[w_a, w_b] = [-10.59, -3.21]$. Furthermore, discounting m at rate $\varepsilon_v = 0.05$ and approximating as the closed interval $[w_c - 100\sigma_w, w_c + 100\sigma_w] = [-129.7, 115.7]$, we can construct the evidence (\mathcal{F}_k^w, m_k^w) of w_k shown in Table 2.

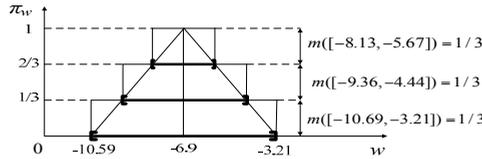


Figure 9. Probability distribution π_w of observation noise w .

Table 2. Evidence of observation noise.

\mathcal{F}_k^w	$[-8.13, -5.67]$	$[-9.36, -4.44]$	$[-10.59, -3.21]$	$[-129.7, 115.7]$
m_k^w	0.3167	0.3167	0.3167	0.05

From Figure 7, it can be seen that the first observation value of resonance frequency $z_1 = 1023.3$ Hz. According to Step 2) in Section 3.2, the first estimation result $\hat{x}_{1|1}$ is initialized as the real observation z_1 . After obtaining (\mathcal{F}_1^v, m_1^v) and (\mathcal{F}_1^w, m_1^w) , our recursive algorithm presented in Section 3.2 can be used to estimate resonance frequencies at each step k . Figure 10a gives the estimation results of our method and Nassreddine’s method, together with the true values and observations (z_k). Figure 10b gives the absolute errors between true values and the estimated values of our method, the estimated values of Nassreddine’s method, and z_k respectively. It can be seen that the estimation accuracy and convergence of our method are better than those of Nassreddine’s method because of the focusing function of the proposed fusion procedure for dependent evidence.

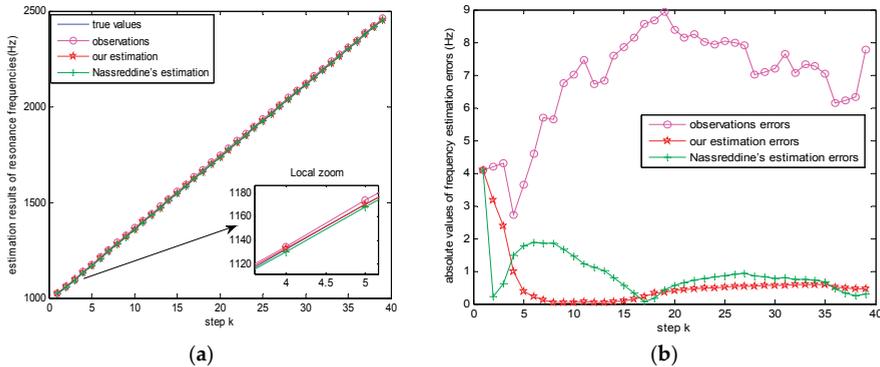


Figure 10. (a) Estimation results of resonance frequencies, (b) Absolute values of frequency estimation errors.

Finally, we can calculate the estimate level L by Equation (32) according to the estimated resonance frequencies of our method, Nassreddine’s method, and z_k respectively as shown in Figure 11a,b gives the corresponding absolute values of length estimation errors. Obviously, the more accurate the estimations of resonance frequencies are, the more accurate the estimations of level L are. As our method always provides more accurate estimations of resonance frequencies, it is therefore always superior.

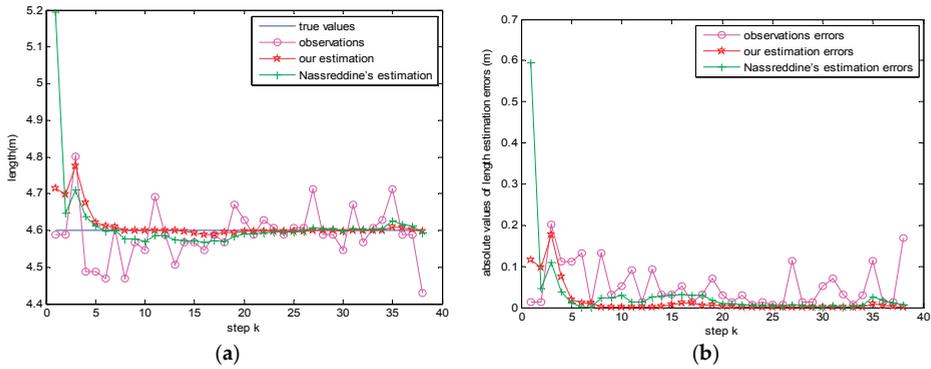


Figure 11. (a) Estimation results of level L , (b) Absolute values of length estimation errors.

More experiments are performed for different values of L to find the mean of absolute values of estimation errors and to show the efficacy of the proposed method as shown in Table 3. Here, for every values of L , from above to below, Table 3 gives in order the experiment results of our method, Nassreddine’s method, and direct measurement method (namely, substituting z_k into (32)).

Table 3. Experimental results for different values of L .

No	True L (m)	T (°C)	Runtime (s)	Mean Error (m)	No	True L (m)	T (°C)	Runtime (s)	Mean Error (m)
1	1.3	27	1.81	0.0126	6	5.6	26.5	20.11	0.018
			0.88	0.016				4.81	0.0374
			-	0.238				-	0.0661
2	2.1	26.5	2.49	0.0254	7	6.6	26.5	23.57	0.0238
			1.33	0.0364				5.61	0.0436
			-	0.0441				-	0.088
3	2.6	26.5	7.81	0.0144	8	7.6	26.5	27.94	0.0299
			2.05	0.0297				6.77	0.0530
			-	0.0591				-	0.1060
4	3.6	26.5	9.62	0.0141	9	8.6	23.9	31.23	0.0216
			2.34	0.0312				7.41	0.0456
			-	0.0468				-	0.1295
5	4.6	26.5	16.07	0.0160	10	9.6	23.9	35.15	0.0435
			3.81	0.0337				8.36	0.0732
			-	0.0552				-	0.1624

It should be noted that the calculation complexity of our algorithm is relatively high, and meanwhile, with the increase of the measured length of level, the corresponding synthetic wave contains more and more resonance frequency points, so the CPU time will increase, so it needs a longer time (the hardware in this test: CPU E8400, CPU Clock Speed 3.00 GHz, RAM 2 GB). But to the situation that liquid level change relatively slowly, our method is applicable. Certainly, the rapid development of data processing capability of computer hardware will make the complexity less of an issue.

5. Conclusions

The subject of Nassreddine’s method is still interval analysis, it introduces evidence, namely belief-function to elaborate bounded noises. In detail, it gives the improved form of noise bounds (the triangular possibility distribution), here the error interval is extended to the evidence construction, namely many interval focal elements with the corresponding belief assignments. Obviously, the latter has more information than the former. Then, it still uses interval arithmetic and constraint-satisfaction techniques to propagate not only interval focal elements, but also its belief assignments, hence its

performance is slightly better than that of pure interval propagation. However, the subject of our method is DS evidence theory and random set theory and introduces Dempster combination rule and evidence dependence conceptions. Although we still use Nassreddine’s evidence construction technique, the random set description of evidence and extension principle of random set are used to obtain state evidence and observation evidence from the defined information sources and to propagate them in the system equations. The main contribution is to realize the fusion of the propagated evidence, in which the degree of dependence and the combination of dependent evidence are further considered.

As a whole, compared with Nassreddine’s method, our method increases the state estimation accuracy. The application in liquid level estimation using an industrial level apparatus shows the efficacy of the proposed method. Certainly, it is worth noting that, in a given application, there are some constraint conditions such as the continuous, monotonic and invertible properties of state and observation equations, and state observability. When they cannot be satisfied, the computational burden will inevitably be increased because of the use of additional complex interval operation algorithms or matrix operation algorithms (for multidimensional states) in [21,22]. Hence fast operation algorithms should be studied in the further. On the other hand, although the proposed procedure of evidence fusion can make the masses focus to those interval focal elements that contain the system state, so as to get the accurate estimation results, how to further evaluate the convergence of fusion using available theories is still a problem worthy studying which will promote the usage of evidence theory in state estimation.

Acknowledgments: This work was supported by the NSFC (No. 61433001, 61374123, 61573275, U1509203), and the Zhejiang Province Research Program Project of Commonweal Technology Application (No. 2016C31071).

Author Contributions: Xiaobin Xu and Zhenghui Li conceived and designed the experiments; Zhenghui Li performed the experiments; Guo Li and Zhenghui Li analyzed the data; Zhe Zhou contributed materials and analysis tools; Xiaobin Xu wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

We first generalize the definition of evidence energy as shown in Proposition A1.

Proposition A1. Suppose $E = (\mathcal{F}_x, m)$ is a body of evidence whose focal elements are the closed intervals. Energy of the evidence E can be defined by:

$$En'(E) = \sum_{i=1}^{n(E)} \frac{m([x_i^-, x_i^+])}{d([x_i^-, x_i^+]) / \min(d([x_i^-, x_i^+]))} \quad [x_i^-, x_i^+] \neq \Theta \tag{A1}$$

where $[x_i^+, x_i^-]$ denotes interval focal element, $d([x_i^-, x_i^+])$ means interval width and $n(E)$ is the number of interval focal elements. For example, if $E = (\mathcal{F}_x, m)$, $A_1 = [-0.3, 2.6]$, $m(A_1) = 0.3$; $A_2 = [0.3, 1.9]$, $m(A_2) = 0.7$, $\min d([x_i^-, x_i^+]) = 1.6$, then:

$$En'(E) = \frac{0.3}{2.9/1.6} + \frac{0.7}{1.6/1.6} = 0.8655 \tag{A2}$$

It can be proved that $En'(E)$ meets the three conditions of the evidence energy: (1) if $m(\Theta) = 1$, then $En'(E) = 0$ and E represents no useful information; (2) if $d([x_i^-, x_i^+]) = \min_i(d([x_i^-, x_i^+]))$ and $m(\Theta) = 0$, then $En'(E) = 1$ and E contains the maximum useful information; (3) $En'(E) \in [0, 1]$.

Proof. For mass function satisfying $\sum_{[x_i^-, x_i^+] \subseteq \Theta} m([x_i^-, x_i^+]) = 1$ and $d([x_i^-, x_i^+]) / \min(d([x_i^-, x_i^+])) \geq 1$, there are four cases to discuss:

Case 1: If $m(\Theta) = 1$ and the mass of other focal elements is zero, then $En'(E) = 0$, namely, the evidence does not provide any information. So, the Condition (1) holds;

Case 2: If $m(\Theta) = 0$ and any interval focal element $[x_i^+, x_i^-]$ meets that $d([x_i^-, x_i^+]) / \min_i(d([x_i^-, x_i^+])) = 1$, then $En'(E) = 1$, namely, the evidence provides maximum useful information. The Condition (2) holds;

Case 3: If $m(\Theta) = 0$ and not all of the interval focal elements $[x_i^+, x_i^-]$ meet that $d([x_i^-, x_i^+]) / \min_i(d([x_i^-, x_i^+])) = 1$, then $En'(E) \in (0, 1)$;

Case 4: If $0 < m(\Theta) < 1$, then $En'(E) \in (0, 1)$;

From Case 1 to Case 4, the Conditions (3) can be proved. Let $E_1 = (\mathcal{F}_x, m_x)$ and $E_2 = (\mathcal{F}_y, m_y)$ be two pieces of evidence with closed interval focal elements. If some focal elements of E_1 and E_2 be induced by the same information sources, then the energy of the intersection of the two pieces of evidence can be described by:

$$En'(E_1, E_2) = \sum_{\substack{ij = 1 \\ D_{ij} \neq \emptyset}}^{|\{D_{ij}\}|} \frac{m(D_{ij})}{d(D_{ij}) / \min\{\min_{x_i} \{d([x_i^-, x_i^+])\}, \min_{y_j} \{d([y_j^-, y_j^+])\}\}} \tag{A3}$$

where D_{ij} is the focal elements induced by the same sources, $\{D_{ij}\}$ is the set of these focal element, $|\{D_{ij}\}|$ is the number of them. Consequentially, the degree of dependence between E_1 and E_2 can be obtained by Definition 5:

$$D'(E_1, E_2) = \frac{2En'(E_1, E_2)}{En'(E_1) + En'(E_2)} \tag{A4}$$

and the dependency coefficient between E_1 and E_2 can be redefined as:

$$R'_{12} = \frac{1}{2} D'(E_1, E_2) \frac{En'(E_2)}{En'(E_1)} \tag{A5}$$

$$R'_{21} = \frac{1}{2} D'(E_1, E_2) \frac{En'(E_1)}{En'(E_2)} \tag{A6}$$

E_1 and E_2 can be modified by R'_{12} and R'_{21} respectively to obtain the corresponding two pieces of independent evidence E'_1 and E'_2 , their BBA functions are given by:

$$m'_1([x_i^-, x_i^+]) = \begin{cases} m_1([x_i^-, x_i^+]) (1 - R'_{12}), & [x_i^-, x_i^+] \neq \Theta \\ 1 - \sum_{[x_i^-, x_i^+] \subset \Theta} m'_1([x_i^-, x_i^+]) & [x_i^-, x_i^+] = \Theta \end{cases} \tag{A7}$$

$$m'_2([y_j^-, y_j^+]) = \begin{cases} m_2([y_j^-, y_j^+]) (1 - R'_{21}), & [y_j^-, y_j^+] \neq \Theta \\ 1 - \sum_{[y_j^-, y_j^+] \subset \Theta} m'_2([y_j^-, y_j^+]) & [y_j^-, y_j^+] = \Theta \end{cases} \tag{A8}$$

After obtaining BBAs of E'_1 and E'_2 , we can use Dempster combination rule to fuse E'_1 and E'_2 , namely, to fuse the dependent evidence E_1 and E_2 indirectly. □

Although Wu, Yang and Liu [17] gave the definition of the energy of the intersection as in Equation (3), this conception of intersection is obscure, namely, D_{ij} and $m(D_{ij})$ are rarely clearly defined. Nevertheless, in our method, the two pieces of dependent evidence are $(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x)$ and $(\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x)$, according to the interval operations (extension principles and combination rule) used to generate them, $\{D_{ij}\} = \hat{\mathcal{R}}_{k+1}^x \cap \mathcal{R}_{k+1|k}^x$ and $m(D_{ij}) = \hat{\rho}_{k+1}^x(D_{ij}) = \rho_{k+1|k}^x(D_{ij})$.

For example, if the focal elements of the new fusion evidence $(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x)$ can be given as: $A_1 = [-0.30, 2.60], m(A_1) = 0.3; A_2 = [0.30, 1.90], m(A_2) = 0.6, A_3 = [0.32, 1.93], m(A_3) = 0.1$; the focal elements of the state estimation evidence $(\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x)$ can be given as: $A_1 = [0.21, 3.50], m(A_1) = 0.1, A_2 = [0.41, 1.61], m(A_2) = 0.3, A_3 = [0.30, 1.90], m(A_3) = 0.6$, then $D_{ij} = [0.30, 1.90], m(D_{ij}) = 0.6$. From

Equation (A1), obviously, we can get $\text{En}'(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x) = 0.8655$, $\text{En}'(\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x) = 0.7865$, then the energy of the intersection of the two pieces of evidence can be calculated by Equation (A3):

$$\text{En}'(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x), (\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x) = \frac{m([0.3, 1.9])}{d([0.3, 1.9]/\min\{1.6, 1.2\})} = \frac{0.6}{1.6/1.2} = 0.45 \quad (\text{A9})$$

We can calculate the degree of dependence between the new fusion evidence and the state estimation evidence by Definition 5 and from Equation (A4), we can get:

$$D'((\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x), (\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x)) = \frac{2 \times \text{En}'((\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x), (\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x))}{\text{En}'(\hat{\mathcal{R}}_{k+1}^x, \hat{\rho}_{k+1}^x) + \text{En}'(\mathcal{R}_{k+1|k}^x, \rho_{k+1|k}^x)} = \frac{2 \times 0.45}{0.8655 + 0.7865} = 0.5448 \quad (\text{A10})$$

The new fusion evidence is denoted as E_1 , and the state estimation evidence is denoted as E_2 . From Equations (A5) and (A6), we can get the dependency coefficient between E_1 and E_2 :

$$R'_{12} = \frac{1}{2} \times 0.5448 \times \frac{0.7865}{0.8655} = 0.2475, \quad R'_{21} = \frac{1}{2} \times 0.5448 \times \frac{0.8655}{0.7865} = 0.2997 \quad (\text{A11})$$

From Equations (A7) and (A8), we can calculate the BBAs of the corresponding E_1' and E_2' as $m'_1(A1) = 0.4733$, $m'_1(A2) = 0.4515$, $m'_1(A3) = 0.0752$ and $m'_2(A1) = 0.3697$, $m'_2(A2) = 0.2101$, $m'_2(A3) = 0.4202$. Finally, using the Dempster combination rule to fuse E_1' and E_2' , we obtain state estimation evidence ($\hat{\mathcal{F}}_{k+1|k+1}^x, \hat{m}_{k+1|k+1}^x$) as: $A_1 = [0.21, 2.60]$, $m(A_1) = 0.1749$, $A_2 = [0.41, 1.61]$, $m(A_2) = 0.0994$, $A_3 = [0.30, 1.90]$, $m(A_3) = 0.1988$, $A_4 = [0.30, 1.90]$, $m(A_4) = 0.1669$, $A_5 = [0.41, 1.61]$, $m(A_5) = 0.0948$, $A_6 = [0.30, 1.90]$, $m(A_6) = 0.1890$, $A_7 = [0.32, 1.93]$, $m(A_7) = 0.0278$, $A_8 = [0.41, 1.61]$, $m(A_8) = 0.0157$, $A_9 = [0.32, 1.90]$, $m(A_9) = 0.0315$.

References

1. Song, Y.; Nuske, S.; Scherer, S. A Multi-Sensor Fusion MAV State Estimation from Long-Range Stereo, IMU, GPS and Barometric Sensors. *Sensors* **2017**, *17*, 11. [CrossRef] [PubMed]
2. Gao, S.; Liu, Y.; Wang, J. The Joint Adaptive Kalman Filter (JAKF) for Vehicle Motion State Estimation. *Sensors* **2016**, *16*, 1103. [CrossRef] [PubMed]
3. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
4. Kalman, R.E.; Bucy, R.S. New Results in Linear Filtering and Prediction Theory. *J. Basic Eng.* **1961**, *83*, 95–108. [CrossRef]
5. Sunahara, Y.; Yamashita, K. An approximate method of state estimation for non-linear dynamical systems with state-dependent noise. *Int. J. Control.* **1970**, *11*, 957–972. [CrossRef]
6. Gordon, N.J.; Salmond, D.J.; Smith, A.F.M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Proc. Inst. Elect. Eng. F* **1993**, *140*, 107–113. [CrossRef]
7. Nassreddine, G.; Abdallah, F.; Denoux, T. State estimation using interval analysis and belief-function theory: application to dynamic vehicle localization. *IEEE Trans. Syst. Man Cybern. B Cybern.* **2010**, *40*, 1205–1218. [CrossRef] [PubMed]
8. Brynjarsdóttir, J.; O'Hagan, A. Learning about physical parameters: The importance of model discrepancy. *Inverse Prob.* **2014**, *30*, 114007. [CrossRef]
9. Bertsekas, D.P.; Rhodes, I.B. Recursive state estimation for a set-membership description of uncertainty. *IEEE Trans. Autom. Control.* **1971**, *16*, 117–128. [CrossRef]
10. Maksarov, D.; Norton, J.P. State bounding with ellipsoidal set description of the uncertainty. *Int. J. Control.* **1996**, *65*, 847–866. [CrossRef]
11. Servi, L.; Ho, Y. Recursive estimation in the presence of uniformly distributed measurement noise. *IEEE Trans. Autom. Control* **1981**, *26*, 563–565. [CrossRef]
12. Gning, A.; Bonnfait, P. Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica* **2006**, *42*, 1167–1175. [CrossRef]

13. Khemane, F.; Abbas-Turki, M.; Durieu, C.; Raynaud, H.F.; Conan, J.M. Bounded-error state and parameter estimation of tip-tilt disturbances in adaptive optics systems. *Int. J. Adapt Control Signal Process.* **2014**, *28*, 1081–1093. [CrossRef]
14. Jaulin, L. Range-only slam with occupancy maps: A set-membership approach. *IEEE Trans. Robot* **2011**, *27*, 1004–1010. [CrossRef]
15. Dempster, A.P. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339. [CrossRef]
16. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
17. Wu, Y.; Yang, J.; Liu, K.; Liu, L. On the evidence inference theory. *Inform Sci.* **1996**, *89*, 245–260. [CrossRef]
18. Dubois, D.; Prade, H. Random sets and fuzzy interval analysis. *Fuzzy Sets Syst.* **1991**, *42*, 87–101. [CrossRef]
19. Xu, X.; Zhou, D.; Ji, Y.; Wen, C. Approximating Probability Distribution of Circuit Performance Function for Parametric Yield Estimation Using Transferable Belief Model. *Sci. Chin. Inf. Sci.* **2013**, *56*, 50–69. [CrossRef]
20. Goodman, I.R.; Mahler, R.P.S.; Nguyen, H.T. *Mathematics of Data Fusion*; Springer: Houten, The Netherlands, 1997.
21. Jaulin, L.; Kieffer, M.; Didrit, O.; Walter, E. *Applied Interval Analysis*; Springer: London, UK, 2001.
22. Dong, W.; Shah, H.C. Vertex method for computing functions of fuzzy variables. *Fuzzy Sets Syst.* **1987**, *42*, 65–78. [CrossRef]
23. Smets, P. Belief functions on real numbers. *Int. J. Approx. Reason.* **2005**, *40*, 181–223. [CrossRef]
24. Kazys, R.; Sliteris, R.; Rekuviene, R. Ultrasonic technique for density measurement of liquids in extreme conditions. *Sensors* **2015**, *15*, 19393–19415. [CrossRef] [PubMed]
25. Donlagić, D.; Kumperščak, V.; Završnik, M. Low-frequency acoustic resonance level gauge. *Sens. Actuators A Phys.* **1996**, *57*, 209–215. [CrossRef]
26. Donlagić, D.; Završnik, M.; Sirotic, I. The use of one-dimensional acoustical gas resonator for fluid level measurements. *IEEE Trans. Instrum Meas.* **2000**, *49*, 1095–1100. [CrossRef]
27. Xu, X.B.; Zhang, Z.; Zheng, J. State Estimation Method Based on Evidential Reasoning Rule. In Proceedings of the 2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 19–20 December 2015.
28. Xu, X.B.; Zhao, C.P.; Xia, B.D.; Wang, Z.; Wen, C.L. A fluid level measurement method based on acoustical resonance in a special frequency range. *Acta Metroa Sin.* **2011**, *32*, 53–57.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Auxiliary Truncated Unscented Kalman Filtering for Bearings-Only Maneuvering Target Tracking

Liang-Qun Li *, Xiao-Li Wang, Zong-Xiang Liu and Wei-Xin Xie

Automatic Target Recognition Key Laboratory (ATR), Shenzhen University, Shenzhen 518060, China; wangxiaoli2016@email.szu.edu.cn (X.-L.W.); liuzx@szu.edu.cn (Z.-X.L.); wxxie@szu.edu.cn (W.-X.X.)

* Correspondence: lqli@szu.edu.cn

Academic Editors: Shuli Sun, Hong Wei and Feng-Bao Yang

Received: 13 March 2017; Accepted: 25 April 2017; Published: 27 April 2017

Abstract: Novel auxiliary truncated unscented Kalman filtering (ATUKF) is proposed for bearings-only maneuvering target tracking in this paper. In the proposed algorithm, to deal with arbitrary changes in motion models, a modified prior probability density function (PDF) is derived based on some auxiliary target characteristics and current measurements. Then, the modified prior PDF is approximated as a Gaussian density by using the statistical linear regression (SLR) to estimate the mean and covariance. In order to track bearings-only maneuvering target, the posterior PDF is jointly estimated based on the prior probability density function and the modified prior probability density function, and a practical algorithm is developed. Finally, compared with other nonlinear filtering approaches, the experimental results of the proposed algorithm show a significant improvement for both the univariate nonstationary growth model (UNGM) case and bearings-only target tracking case.

Keywords: bearings-only target tracking; statistical linear regression; auxiliary truncated unscented Kalman filtering

1. Introduction

Bearings-only maneuvering target tracking has been widely researched for decades. It is important for many applications such as maritime surveillance, navigation and aerospace, wireless sensor networks (WSN), and infrared search and track (IRST) systems [1–6]. However, while implementing this technology in unlimited situations, there remain some challenging problems, such as multiple platform tracking, uncertainty of the target model and nonlinear non-Gaussian noise. To deal with the uncertainty of the motion model, such as abrupt target maneuver, heavy clutter measurements, highly nonlinearity of dynamic models and nonlinear non-Gaussian noise, etc., the interacting multiple model (IMM) [7] based on the nonlinear filtering algorithm is a promising approach. However, to model the uncertainty of the motion model, the performance of the IMM-type algorithm is directly proportional to the number of the motion models. Generally, the more motion models we produce, the greater accuracy of the estimated state we obtain. However, the computational complexity of the algorithm becomes larger with the increase of the numbers of motion models, particularly in heavily cluttered environments. Moreover, the nonlinear filtering has been studied extensively in bearings-only maneuvering target tracking.

As is well-known, the most widely used nonlinear filtering for bearing-only tracking is to employ an extended Kalman filter (EKF) [8,9]. However, when the nonlinearity of dynamic models becomes more severe, the performance of the EKF degrades sharply. In order to solve this problem, the unscented Kalman filter (UKF) [10] and the truncated unscented Kalman filtering (TUKF) were proposed [11,12]. Compared with other conventional Kalman filter-type approaches, the TUKF can achieve better performance in the conditions of the target tracking system, and can provide very

informative nonlinear measurements compared to the prior. Moreover, to take into account the available additional information the state given by the constraint, Ondrej et al. [13] proposed a generic local filter for the inequality constrained estimation problem, and designed an efficient truncation technique based on the Monte Carlo integration method for the approximation of the state probability density function. Beatriz et al. [14] proposed a constrained dual state and parameter estimation algorithm using a dual Kalman filter (DKF) and a probability density function (PDF) truncation algorithm for analysis of lateral vehicle dynamics.

In recent years, particle filtering has been widely used for bearing-only tracking. In [15], Gordon proposed the first particle filtering algorithm based on the resampling step. The main idea is that the posterior distribution can be approximated by series of random samples with associated weights, and its parameter estimates can be computed by these samples and weights. Therefore, particle filtering can deal with nonlinear non-Gaussian problems in terms of the dynamics and measurements. Recently, many particle filtering methods have been proposed [16–18], for example, the extended Kalman particle filter (EKF-PF), unscented particle filter (UPF), and the multivariable feedback particle filter (GPF) [18]. Moreover, for the maneuvering target tracking problem, many particle filters have been proposed based on Markovian switching systems [19–22]. Boers et al. [19] proposed an interacting multiple model particle filter algorithm (IMM-PF) by combining a mixture of the interacting multiple model (IMM) filter with the particle filter. For the maneuvering target tracking problem in bearings-only wireless sensor networks (WSNs), Atiyeh et al. [20] proposed an interacting multiple model particle filter to estimate the state variables of the moving target. Li et al. [21] proposed a Rao–Blackwellized particle filter based on multiple model algorithm for maneuvering target tracking in a cluttered environment. Yu et al. [22] proposed a distributed particle filter by incorporating the curvature of the sensing region in the measurement model for bearings-only tracking of a moving target. In their method, to reduce the communication load, the transformation of the observations is approximated as Gaussian distribution, which the variance can be approximated using the average variance over all particles. However, abrupt target maneuvers, modeling uncertainty and the high nonlinearity of model function remain to be unsolved issues.

For achieving a successful tracking performance, the aforementioned methods require accurate motion models and adaptive nonlinear filtering methods. However, particularly in maneuvering target tracking, accurate motion modeling is almost impossible, and an adaptive nonlinear filtering needs to be used to handle abrupt maneuver of target. More importantly, these two challenges are not separate problems and should be considered simultaneously. In previous research [23,24], Ehsan et al. [23] proposed a new bearing-only bias estimation model based on triangulation using the associated measurement reports (AMR) or local bearing-only tracks from different sensor pairs for distributed tracking systems. Li et al. [24] proposed novel truncated quadrature Kalman filtering (TQKF) based on the Gauss-Hermite quadrature rule for bearings-only maneuvering target tracking. In order to avoid the requirement of the measurement function being bijective, the modified prior PDF of the TQKF algorithm can be approximately computed by the least square estimation approach. However, the most important limitation of the TQKF is the expensive computational burden, and it cannot be used for real time target tracking.

In this paper, novel auxiliary truncated Kalman filtering (ATUKF) is proposed for bearings-only maneuvering target tracking. Unlike the TUKF algorithm, to overcome the modeling uncertainty, a modified prior probability density function (PDF) is defined based on several auxiliary target characteristics and current measurements, which can effectively minimize the variance of the prior distribution. Moreover, to achieve the requirement of bijective measurement function, the statistical linear regression based on the unscented transformation is used to linearize the nonlinear measurement function, and the modified prior PDF is approximated as Gaussian. Finally, the posterior PDF can be approximately estimated based on the prior PDF and the modified prior PDF, and a practical algorithm is developed for bearings-only target tracking systems.

The rest of the paper is organized as follows. The proposed algorithm is given in Section 2. In Section 3, we provide the experimental results. Finally, some conclusions are given in Section 4.

2. Proposed Algorithm

In order to track the maneuvering target, accurate motion modeling and nonlinear filtering are two challenging problems that should not be separated. However, most research on maneuvering tracking investigates these problems separately. In this section, in the bearings-only maneuvering target tracking, novel auxiliary truncated unscented Kalman filtering is proposed. In Section 2.1, the joint prior distribution is approximately constructed. In Section 2.2, in order to track the bearings-only maneuvering target, the modified prior PDF is approximated based on statistical linear regression by introducing the target spatio-temporal information. Section 2.3 summarizes the proposed algorithm.

2.1. Joint Prior Distribution

Suppose the target dynamic system can be written as:

$$x_n = f(x_{n-1}) + m_n \tag{1}$$

$$z_n = h(x_n) + e_n \tag{2}$$

where $z_n \in R^{n_z}$ denotes the observation vector at time n , $x_n \in R^{n_x}$ denotes the target state vector at time n , and $f(\cdot)$ and $h(\cdot)$ denote the nonlinear state transition function and observation state function, respectively.

Suppose that r_n denotes the set of target characteristics including c independent components $r_n = \{r_n^1, r_n^2, \dots, r_n^c\}$. In order to derive the proposed algorithm, there are two basic hypotheses, firstly, that the nonlinear function $h_n(\cdot)$ in (2) is a bijective, continuous function; and secondly, that the probability density function of the measurement noise e_n has bounded, connected support.

$$p_{e_n}(v_n) = 0, e_n \notin I_{e_n} \subset \mathfrak{R}^{n_z} \tag{3}$$

where I_{e_n} is an n_z -dimensional measurement validation region. Therefore, according to the second assumption, the measurement likelihood function can be defined as follows:

$$p(z_n|x_n, r_n) = p_{e_n}(z_n - h_n(x_n))\chi_{I_{e_n}}(z_n - h_n(x_n)) \tag{4}$$

$$p(z_n|x_n, r_n) = p_{e_n}(z_n - h_n(x_n))\chi_{I_{x_n}(z_n)}(x_n) \tag{5}$$

$$I_{x_n}(z_n) = \left\{ x_n | x_n = h_n^{-1}(z_n - e_n), e_n \in I_{e_n} \right\} \tag{6}$$

where $\chi_{I_{e_n}}(\cdot)$ is the indicator function on the subset I_{e_n} . Therefore, the state posterior PDF can be defined as:

$$\begin{aligned} p(x_{0:n}|z_{1:n}, r_{1:n}) &= \frac{p(z_n|x_{0:n}, z_{1:n-1}, r_{1:n})p(x_n|x_{1:n-1}, z_{1:n-1}, r_{1:n})p(x_{1:n-1}|z_{1:n-1}, r_{1:n-1})}{p(z_n, r_n|z_{1:n-1}, r_{1:n-1})} \\ &= \frac{p_{e_n}(z_n - h(x_n))\chi_{I_x(z_n)}(x_n)p(x_n|x_{n-1}, z_{1:n-1}, r_{1:n})p(x_{1:n-1}|z_{1:n-1}, r_{1:n-1})}{p(z_k, r_k|z_{1:k-1}, r_{1:k-1})} \\ &\propto p(z_n|x_n)p_1(x_n|z_n, x_{n-1}, r_{1:n})p(x_{1:n-1}|z_{1:n-1}, r_{1:n-1}) \end{aligned} \tag{7}$$

$$p_1(x_n|z_n, x_{n-1}, r_{1:n}) = p(x_n|x_{n-1}, z_{1:n-1}, r_{1:n})\chi_{I_{x_n}(z_n)}(x_n)/\varepsilon_1 \tag{8}$$

where ε_1 is a constant. From Equation (8), we can see that the modified prior PDF is defined by incorporating the current measurement information z_n . According to the conclusions in [11], if the measurement noise is informative, the modified prior $p_1(\cdot)$ is not only the minimum variance of the prior $p_0(\cdot)$, but also can improve the algorithm's performance. Further, to deal with the uncertainty

of motion models, the joint prior distribution $p(x_n|x_{0:n-1}, z_{1:n}, r_{1:n})$ of the proposed algorithm can be defined as follows:

$$\begin{aligned}
 p(x_n|x_{0:n-1}, z_{1:n}, r_{1:n}) &= \alpha_n p_1(x_n|z_n, x_{n-1}, r_{1:n}) + (1 - \alpha_n) p_0(x_n|x_{n-1}, z_{1:n-1}) \\
 &= \alpha_n p(x_n|x_{n-1}, r_{1:n}) \chi_{I_{x_n}(z_n)}(x_n) + (1 - \alpha_n) p_0(x_n|x_{n-1}, z_{1:n-1})
 \end{aligned} \tag{9}$$

where $\alpha_n \in [0, 1]$ is a proper parameter. To approximately calculate the mean and covariance of the posterior distribution, we apply a UKF update to $p_0(\cdot)$ (UKF1), and another UKF update to $p_1(\cdot)$ (UKF2). Finally, the posterior estimates can be approximately calculated through merging both results obtained by UKF1 and UKF2.

2.2. Approximation of $p_1(\cdot)$

In the subsection, our object is to approximate the modified prior PDF $p_1(\cdot)$ as Gaussian. For this reason, we write the state vector as $x_n = [a_n^T, b_n^T]^T$, where $a_n \in \mathbb{R}^{n_a}$ denotes the position components of x_n , $b_n \in \mathbb{R}^{n_b}$ denotes the velocity components of x_n , and $n_x = n_a + n_b$. The derivation of the mean $\hat{x}_{p_1,n}$ and covariance $P_{p_1,n}$ of $p_1(\cdot)$ is the same as in the truncated unscented Kalman filter (TUKF) [11], which can be shown as follows:

$$\hat{x}_{p_1,n} = \begin{bmatrix} \mu_{a_n,1} \\ \mu_{b_n,1} \end{bmatrix}, P_{p_1,n} = \begin{bmatrix} \Sigma_{a_n,1} & \Sigma_{a_n b_n,1} \\ (\Sigma_{a_n b_n,1})^T & \Sigma_{b_n,1} \end{bmatrix} \tag{10}$$

$$\Sigma_{a_n,1} = \tilde{H}_n^{-1} R_n (\tilde{H}_n^{-1})^T \tag{11}$$

where $\mu_{b_n,1}$, $\Sigma_{b_n,1}$, $\Sigma_{a_n b_n,1}$ can be found in [11]. $\mu_{a_n,1}$ denotes the estimated mean of state a_n , R_n denotes the measurement noise covariance, and $\tilde{H}_n^{-1} = [\nabla_{a_n} h_n^T(a_n)]^T|_{a_n=\mu_{a_n,1}}$ is the Jacobian of $h_n^T(a_n)$ evaluated at $\mu_{a_n,1}$.

Now, how to calculate the estimated mean $\mu_{a_n,1}$ remains a key problem to be solved. For the passive sensor tracking system, the modeling of target dynamic system is a challenging problem when the target maneuvers, and some auxiliary target characteristics need to be used to deal with arbitrary changes in motion models. To achieve a high tracking performance, a statistical linear regression method (SLR) [25] is proposed to estimate the state mean $\mu_{a_n,1}$.

Firstly, to evaluate the state mean $\mu_{a_n,1}$, three approximations are used: (S1) the prior PDF $p_0(a_n)$ is constant over the connected region $I_{a_n}(z_n)$; (S2) the nonlinear function $h(\cdot)$ can be locally linearized; and (S3) the measurement noise satisfies uniform distribution $e_n \sim U_{I_{e_n}}$ in the connected region I_{e_n} . According to S2, the nonlinear measurement Equation (2) can be approximated as a linear estimator of z_n, \hat{z}_n such that:

$$\hat{z}_n = \tilde{h}_n a_n + d_n \tag{12}$$

where \tilde{h}_n denotes a linear measurement matrix, and d_n denotes a noise vector, which are derived by minimizing the objective function defined as follows:

$$\{\tilde{h}_n, d_n\} = \operatorname{argmin} E(\tau_n^T \tau_n) \tag{13}$$

where τ_n is the linearization error, $\tau_n = z_n - \hat{z}_n$.

Substituting τ_n into (13), and setting the partial derivative of the objective function with d_n to zero,

$$\begin{aligned}
 (-2)E(z_n - \tilde{h}_n a_n - d_n) &= 0 \\
 \Leftrightarrow d_n &= \bar{z}_n - \tilde{h}_n \bar{a}_n
 \end{aligned} \tag{14}$$

where $\bar{a}_n = E(x_n)$ and $\bar{z}_n = E(z_n)$. Substituting d_n into (13)

$$\tau^T \tau = [(z_n - \bar{z}_n) - \tilde{h}_n (a_n - \bar{a}_n)]^T [(z_n - \bar{z}_n) - \tilde{h}_n (x_n - \bar{x}_n)]$$

Then, setting the gradient with respect to \hat{h}_n to zero,

$$(-2)E\{[(z_n - \bar{z}_n) - \hat{h}_n(a_n - \bar{a}_n)][z_n - \bar{z}_n]\} = 0 \tag{15}$$

Solving (15) for \hat{h}_n , we obtain:

$$\hat{h}_n = P_{\hat{a}_n z_n}^T P_{\hat{a}_n \hat{a}_n}^{-1} \tag{16}$$

where $P_{\hat{a}_n \hat{a}_n} = E[(a_n - \bar{a}_n)(a_n - \bar{a}_n)^T]$, $P_{\hat{a}_n z_n} = E[(a_n - \bar{a}_n)(z_n - \bar{z}_n)^T]$. According to the measurement equation (12), the maximum likelihood position $\hat{a}_n(z_n)$ of target state a_n can be estimated as follows:

$$\hat{a}(z_n) = \hat{h}_n \cdot (z_n - d_n) \tag{17}$$

If the maximum likelihood estimate $\hat{a}_n(z_n)$ is used to replace the estimated mean $\mu_{a_n,1}$, the performance of the proposed algorithm will be improved because the current measurement information is incorporated. However, it cannot solve the uncertainty in motion models. In particular, when the target speed or the measurement sampling interval is large, the tracking performance degrades.

More recently, sophisticated techniques have been based on the target motion characteristics [6], which have been proposed to address the challenges in motion modeling. In their proposed method, three target characteristics, such as actual target speed v , time interval T of measurement and course angle θ of the target, are considered to improve the tracking performance. In (18), the relationship between three target characteristics and the state predicted error is given:

$$\begin{aligned} \nabla\sigma &= \sqrt{(T \cdot v)^2 + (T \cdot v_n)^2 - 2T^2 v \cdot v_n \cos(\nabla\theta)} \\ &= T \cdot \sqrt{(v)^2 + (v_n)^2 - 2v \cdot v_n \cos(\nabla\theta)} \end{aligned} \tag{18}$$

where v_n denotes the current estimated velocity, $\nabla\theta$ denotes the estimated error of course angle, and v denotes the actual target velocity. Supposing $\nabla v = v - v_n$, we can obtain

$$\begin{aligned} \nabla\sigma &= T \cdot \sqrt{(v)^2 + (v_n)^2 - 2v \cdot v_n \cos(\nabla\theta)} \\ &= T \cdot \sqrt{\nabla v_n^2 + 2v \cdot (\nabla v_n)(1 - \cos(\nabla\theta))} \end{aligned} \tag{19}$$

From Equation (19), we can find that the predicted error $\nabla\sigma$ increases monotonically with the increase of the parameters $(v_n, T, \nabla\theta)$. In fact, when $\nabla v_n > 50$ m/s and $T > 20$ s, the predicted error $\nabla\sigma$ will larger than 1000 m. It shows that the predicted error becomes a major reason for the performance degradation. On the other hand, when the actual target speed is relatively small, the prediction error caused by T or $\nabla\theta$ is smaller than the measurement error. Therefore, to improve the performance of mean estimate $\mu_{a_n,1}$, the maximum likelihood estimate $\hat{a}_n(z_n)$ is considered as the latest observation, and the modified maximum likelihood estimates that can incorporate current target characteristics is defined as follows:

$$\begin{aligned} \hat{\phi}(z_n) &= \mu_{a_n,0} + K_n(\hat{a}(z_n) - \hat{h}_n \mu_{a_n,0}) \\ K_n &= (T^2 \cdot v^2 \cdot \sigma_v^2(n)) / (\lambda \cdot \sigma_e^2(k) + T^2 \cdot v^2 \cdot \sigma_v^2(n)) \end{aligned} \tag{20}$$

where λ is a constant, $\sigma_e^2(n)$ denotes the variance of measurement noise, and $\sigma_v^2(n)$ denotes the innovation variance. According to (20) and (16), the mean $\mu_{a_n,1}$ and the variance $\Sigma_{a_n,1}$ in (10) can be approximated as:

$$\mu_{a_n,1} = \hat{\phi}(z_n), \Sigma_{a_n,1} = P_{\hat{a}_n \hat{a}_n} \tag{21}$$

Finally, the modified prior PDF $p_1(\cdot)$ can be approximated as a Gaussian probability density function $p_1(\cdot) \approx N(x_n, \hat{x}_{p_1,n}, P_{p_1,n})$.

2.3. Summary of the Proposed Algorithm

According to the descriptions above, in order to describe clearly the proposed algorithm, the diagram of the ATUKF is shown in Figure 1. In Figure 1, it is shown that one cycle of the ATUKF algorithm consists of the following steps: (A) time update (predicted by using Kalman filtering); (B) the measurement updates based on the prior $p_0(\cdot)$ and the modified prior $p_1(\cdot)$; and (C) weight calculation and the joint state update. According to the derived results mentioned above and Figure 1, the detailed information of the new ATUKF algorithm can be summarized as follows.

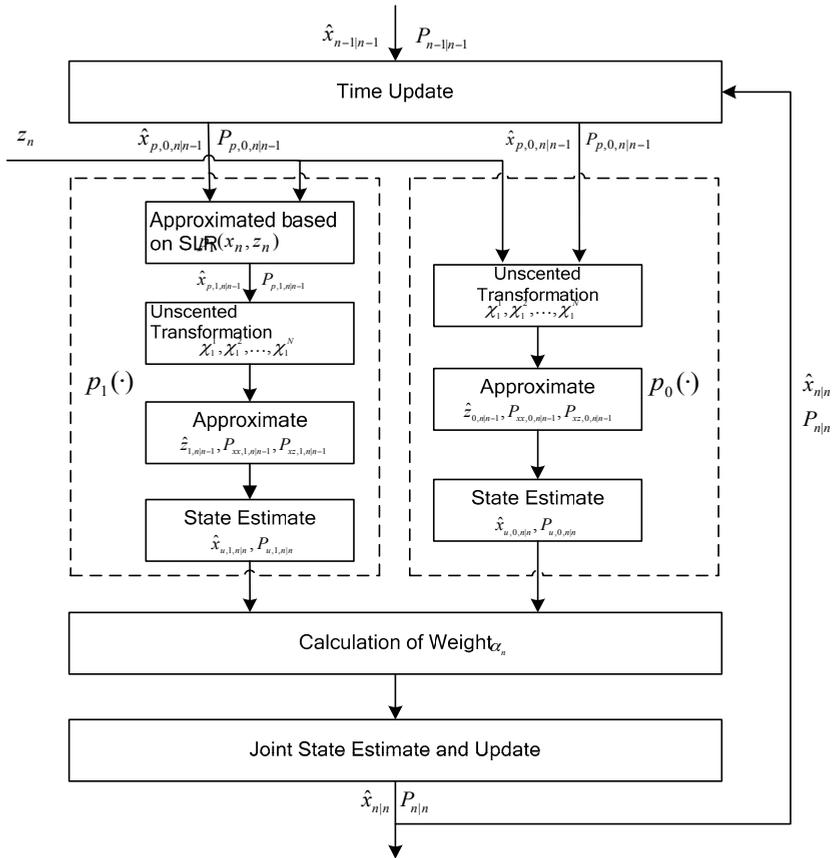


Figure 1. Auxiliary truncated unscented Kalman filtering.

Algorithm: Auxiliary Truncated Unscented Kalman Filtering (ATUKF)

ATUKF—Update based on the prior probability density function (PDF) $p_0(\cdot)$

1. Obtain $N = 2n_x + 1$ sigma points $\chi_0^1, \chi_0^2, \dots, \chi_0^N$ and the corresponding associated weights w_1, w_2, \dots, w_N using unscented transform (UT) based on the mean $\hat{x}_{n-1|n-1}$ and covariance $P_{n-1|n-1}$ of the posterior PDF $p_0(x_n|x_{n-1}, z_{1:n-1}, r_{1:n-1})$, where n_x denotes the dimension of state x . The predicted sigma points can be obtained by the nonlinear state function $f(\cdot)$:

$$\chi_{0,n|n-1}^i = f(\chi_0^i), i = 1, 2, \dots, N \tag{22}$$

2. Approximate the mean and covariance of the state-predicted prior PDF $p_0(x_n|x_{n-1}, z_{1:n-1})$

$$\hat{x}_{p,0,n|n-1} = \sum_{i=1}^N w_i \chi_{0,n|n-1}^i \quad (23)$$

$$P_{p,0,n|n-1} = Q_n + \sum_{i=1}^N w_i (\chi_{0,n|n-1}^i - \hat{x}_{p,0,n|n-1})(\chi_{0,n|n-1}^i - \hat{x}_{p,0,n|n-1})^T \quad (24)$$

3. Compute the predicted measurement $\hat{z}_{0,n|n-1}$ based on the nonlinear measurement function $h(\cdot)$:

$$z_{0,n|n-1}^i = h(\chi_{0,n|n-1}^i), \quad i = 1, 2, \dots, N \quad (25)$$

$$\hat{z}_{0,n|n-1} = \sum_{i=1}^N w_i z_{0,n|n-1}^i \quad (26)$$

4. The cross-covariance, innovation covariance and error covariance are estimated as follows:

$$P_{xz,0,n|n-1} = \sum_{i=1}^N w_i (\chi_{0,n|n-1}^i - \hat{x}_{p,0,n|n-1})(z_{0,n|n-1}^i - \hat{z}_{0,n|n-1})^T, \quad (\text{Cross covariance}) \quad (27)$$

$$P_{zz,0,n|n-1} = R_n + \sum_{i=1}^N w_i (z_{0,n|n-1}^i - \hat{z}_{0,n|n-1})(z_{0,n|n-1}^i - \hat{z}_{0,n|n-1})^T, \quad (\text{Innovation covariance}) \quad (28)$$

$$P_{xx,0,n|n-1} = \sum_{i=1}^N w_i (\chi_{0,n|n-1}^i - \hat{x}_{p,0,n|n-1})(\chi_{0,n|n-1}^i - \hat{x}_{p,0,n|n-1})^T, \quad (\text{Error covariance}) \quad (29)$$

5. Estimate the mean $\hat{x}_{u,0,n|n}$ and covariance $P_{u,0,n|n}$ using (30) and (31):

$$\hat{x}_{u,0,n|n} = \hat{x}_{p,0,n|n-1} + P_{xz,0,n|n-1} P_{zz,0,n|n-1}^{-1} (z_n - \hat{z}_{0,n|n-1}) \quad (30)$$

$$P_{u,0,n|n} = P_{p,0,n|n-1} - P_{xz,0,n|n-1} P_{zz,0,n|n-1}^{-1} P_{xz,0,n|n-1}^T \quad (31)$$

ATUKF—Update based on the modified prior PDF $p_1(\cdot)$

1. Calculation of the mean $\hat{x}_{p,1,n|n-1}$ and covariance $P_{p,1,n|n-1}$ of the prior $p_1(\cdot)$

According to (14) and (16) in Section 2.2, the linear regression coefficients \hat{h}_n and \hat{d}_n can be approximately computed by using Equations (27)–(29). The mean $\hat{x}_{p,1,n|n-1}$ and covariance $P_{p,1,n|n-1}$ of $p_1(\cdot)$ can be approximately estimated by (10) and (11), respectively.

2. Draw N new sigma points $\chi_{1,n|n-1}^1, \chi_{1,n|n-1}^2, \dots, \chi_{1,n|n-1}^N$ with the associated weights w_1, w_2, \dots, w_N by using the UT based on the mean $\hat{x}_{p,1,n|n-1}$ and covariance $P_{p,1,n|n-1}$. The predicted measurements of new sigma points are estimated as follows:

$$z_{1,n|n-1}^i = h(\chi_{1,n|n-1}^i) \quad (32)$$

3. Calculation of $\hat{z}_{1,n|n-1}$, $P_{zz1,n|n-1}$ and $P_{xz1,n|n-1}$

$$\hat{z}_{1,n|n-1} = \sum_{i=1}^N w_i z_{1,n|n-1}^i \quad (33)$$

$$P_{zz,1,n|n-1} = R_n + \sum_{i=1}^N w_i (z_{1,n|n-1}^i - \hat{z}_{1,n|n-1})(z_{1,n|n-1}^i - \hat{z}_{1,n|n-1})^T, \quad (\text{Innovation covariance}) \quad (34)$$

$$P_{xz,1,n|n-1} = \sum_{i=1}^N w_i (\lambda_{1,n|n-1}^i - \hat{x}_{p,1,n|n-1}) (z_{1,n|n-1}^i - \hat{z}_{1,n|n-1})^T, \text{ (Cross covariance)} \quad (35)$$

4. Estimate the mean $\hat{x}_{u,1,n|n}$ and covariance $P_{u,1,n|n}$ using (36) and (37):

$$\hat{x}_{u,1,n|n} = \hat{x}_{p,1,n|n-1} + P_{xz,1,n|n-1} P_{zz,1,n|n-1}^{-1} (z_n - \hat{z}_{1,n|n-1}) \quad (36)$$

$$P_{u,1,n|n} = P_{p,1,n|n} - P_{xz,1,n|n-1} P_{zz,1,n|n-1}^{-1} P_{xz,1,n|n-1}^T \quad (37)$$

ATUKF—Jointly update

1. Calculate the parameter α_n using (38) and (39)

$$\mu_1(\hat{x}_{u,i,n}) = 1/\sqrt{|P_{u,i,n}|} \cdot \exp((z_n - h_n(\hat{x}_{u,i,n}))/2), i = 0, 1 \quad (38)$$

$$\alpha_n = \mu_1(\hat{x}_{u,1,n}) / (\mu_0(\hat{x}_{u,0,n}) + \mu_1(\hat{x}_{u,1,n})) \quad (39)$$

2. Approximate the mean \hat{x}_n and covariance P_n of the posterior PDF $p(x_n|z_{1:n})$ using (23) and (24).

$$\hat{x}_n = \alpha_n \cdot \hat{x}_{u,1,n|n} + (1 - \alpha_n) \cdot \hat{x}_{u,0,n|n} \quad (40)$$

$$P_n = \alpha_n \cdot [P_{u,1,n|n} + (\hat{x}_{u,1,n|n} - \hat{x}_n)(\hat{x}_{u,1,n|n} - \hat{x}_n)^T] + (1 - \alpha_n) \cdot [P_{u,0,n|n} + (\hat{x}_{u,0,n|n} - \hat{x}_n)(\hat{x}_{u,0,n|n} - \hat{x}_n)^T] \quad (41)$$

3. Simulation Results

In this section, to evaluate the tracking performance of the ATUKF algorithm, two examples are employed. In Section 3.1, the univariate nonstationary growth model (UNGM) is discussed [11]. In Section 3.2, a bearings-only maneuvering tracking scenario [24], interested in defense applications, is discussed. In the first case, the EKF, UKF, the quadrature Kalman filtering (QKF) [25], the mixture truncated unscented Kalman filter (MTUKF, with three mixture components) [12] and particle filtering (PF) are utilized. In the second case, the TQKF, interacting multiple model extended Kalman filtering (IIMMEKF) and the interacting multiple model Rao–Blackwellized particle filter (IMMRBPF) are employed. In all the experiments, each simulation has been repeatedly performed 100 times.

3.1. Univariate Nonstationary Growth Model (UNGM)

In this section, due to the highly nonlinearity and non-stationarity of dynamic system, the univariate nonstationary growth model is considered. The discrete time system of this model can be written as:

$$x_n = \alpha x_{n-1} + \beta \frac{x_{n-1}}{1 + x_{n-1}^2} + \gamma \cos(1.2(n - 1)) + m_n \quad (42)$$

$$z_n = \begin{cases} \phi_2 x_n^2 + e_n & n \leq 30 \\ \phi_1 x_n^3 - 2 + e_n & n > 30 \end{cases} \quad (43)$$

where the process noise m_n is satisfied with Gaussian distribution with zero mean and variance one, and e_n is satisfied with Gaussian distribution with zero mean and variance 0.01. $\alpha = 0.5$, $\beta = 25$, $\gamma = 8$, $\phi_1 = 0.2$ and $\phi_2 = 0.05$ are known constants. In each Monte Carlo simulation, we assume that the initial distribution of state x_0 is uniform distribution in the interval [0 1]. The number of particles is 1000.

Figure 2 shows the root-mean-square position errors (RMSE) of the EKF, UKF, QKF, PF, MTUKF and ATUKF. It is obvious from Figure 2 that the performance of the ATUKF is much better than that of the EKF, UKF and QKF. In this case, the performance of the EKF is the poorest. A reason for the poor performance of the EKF, UKF and QKF is the increase of the approximate error arising from the high

nonlinearity and the non-stationarity of the dynamic system. Figure 3 shows the RMS position errors of all filters with different noise variance $\sigma_{e_n} \sim [0.1 \ 5]$. From Figure 3, it is seen that whenever the measurement is informative ($\sigma_{e_n} < 1$) or the measurement is uninformative ($\sigma_{e_n} > 1$), the ATUKF is robust in all situations, its performance is similar to the MTUKF's, and it is very close to that of the PF. Moreover, among the EKF, UKF and QKF, the EKF has the poorest performance. In particular, when the measurement is very informative ($\sigma_{e_n} < 1$), the EKF yields a divergent estimate.

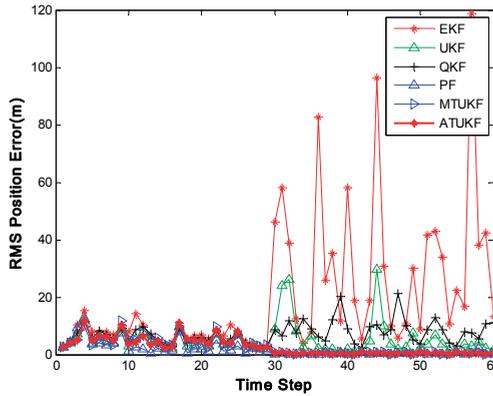


Figure 2. Root-mean-square (RMS) position errors of the extended Kalman filter (EKF), unscented Kalman filter (UKF), quadrature Kalman filtering (QKF), PF, mixture truncated unscented Kalman filter (MTUKF) and auxiliary truncated unscented Kalman filtering (ATUKF).

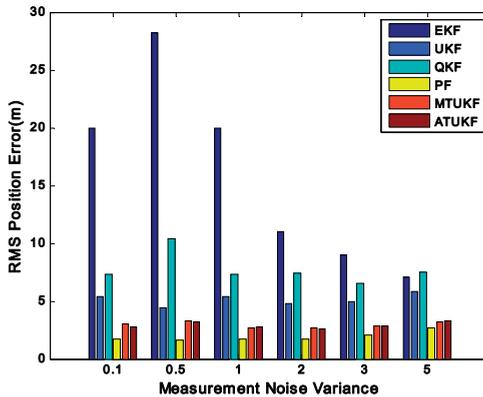


Figure 3. RMS position error for different noise variances.

Table 1 shows the computation time statistics for all algorithms. In this case, all the experiments are performed by using MATLAB programming on Intel-Core(TM)-i2-4030U processor (1.9 GHz) based on the Windows platform. It can be seen from Table 1 that the computational load of the PF is the largest than these of other filters, such as the EKF, UKF, QKF, MTUKF and ATUKF. The ATUKF is very close to the QKF. Furthermore, the computation time for the ATUKF is much lower than the MTUKF. The main reason is that the MTUKF approximates the posterior PDF as a Gaussian mixture, and it makes the computational burden increase.

Table 1. Comparison of the computation times of different filtering algorithms (s). UNGM: univariate nonstationary growth model.

Case	EKF	UKF	QKF	PF	MTUKF(3)	ATUKF
UNGM	1.102	6.650	15.264	522.519	43.142	16.240

3.2. Bearings-Only Maneuvering Tracking (BOT) Scenario

In this scenario, the target makes two circular turns with rectilinear segments connecting them. Figure 4 shows the true target trajectory. The speed modulus is kept constant throughout (0.3 km/s). The initial position is (2 km, 8 km, 1 km), and the initial velocity is (0.15 km/s, 0.26 km/s, 0.0 km/s). The segments are defined as follows:

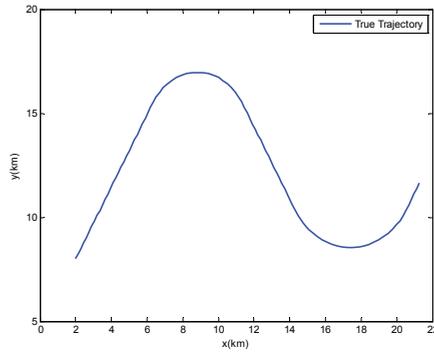


Figure 4. True Trajectory.

First segment. Rectilinear flight until the plane is at (6.35 km, 15.53 km, 1 km) (from $t = 0$ s to $t = 30$ s).

Second segment. Circular turn with turn rate $6^\circ/\text{s}$ (from $t = 31$ s to $t = 50$ s).

Third segment. Rectilinear flight until the plane is at (14.31 km, 10.33 km, 1 km) (from $t = 51$ s to $t = 70$ s).

Fourth segment. Circular turn with turn rate $4.8^\circ/\text{s}$ (from $t = 71$ s to $t = 95$ s).

Fifth segment. Rectilinear flight until the plane is at (21.26 km, 11.63 km, 1 km) (from $t = 96$ s to $t = 100$ s).

The motion model of target is defined as follows:

$$x_n^s = F^s x_{n-1}^s + m_n^s \quad (44)$$

where $x_n^s = (x, y, z, x', y', z')$ denotes the target state vector under model x , x , y and z denote the target position coordinates, x' , y' and z' are the target speed in x , y and z directions, respectively, F^s denotes the transition matrix, and $s \in \{1, 2, \dots, M\}$ denotes the target model index. In the maneuvering target tracking scenario, only a constant velocity model is used for the ATUKF algorithm and TQKF algorithm. In the IMM-EKF and IMM-RBPF, there are both clockwise- and counterclockwise-coordinated turn models that are used to simulate the target maneuvering. The details of three target motion models are defined as follows:

Model 1: Constant Velocity Motion

The state transition matrix and the process noise covariance matrix are defined by:

$$F^1 = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (45)$$

$$Q^1 = \begin{bmatrix} \frac{1}{4}T^4 & 0 & 0 & \frac{1}{2}T^3 & 0 & 0 \\ 0 & \frac{1}{4}T^4 & 0 & 0 & \frac{1}{2}T^3 & 0 \\ 0 & 0 & \frac{1}{4}T^4 & 0 & 0 & \frac{1}{2}T^3 \\ \frac{1}{2}T^3 & 0 & 0 & T^2 & 0 & 0 \\ 0 & \frac{1}{2}T^3 & 0 & 0 & T^2 & 0 \\ 0 & 0 & \frac{1}{2}T^3 & 0 & 0 & T^2 \end{bmatrix} \cdot \sigma_n^2 \quad (46)$$

where T denotes the sampling interval, in this paper, T is set to 1.

Model 2: Constant Turn Motion

The state transition matrix is:

$$F^2 = \begin{bmatrix} 1 & 0 & 0 & \frac{\sin(w)}{w} & \frac{\cos(w)-1}{w} & 0 \\ 0 & 1 & 0 & \frac{1-\cos(w)}{w} & \frac{\sin(w)}{w} & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & \cos(w) & -\sin(w) & 0 \\ 0 & 0 & 0 & \sin(w) & \cos(w) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (47)$$

where w is a constant angular rate. In this paper, w is set to 0.0175. The process noise covariance matrix Q^2 is the same as in Model 1.

Model 3: $w > 0$ describes a clockwise turn, and Model 3 is its natural counterpart for a counterclockwise turn $w < 0$.

Two passive sensors are deployed in $(0, -5 \text{ km}, 0)$ and $(0, 5 \text{ km}, 0)$ respectively. Using the detection fusion architecture [24], the azimuth and elevation angles of aircraft, α_i and β_i , respectively, measured by sensor i , are transmitted to the fusion node. The measurement function is written as:

$$h(x_n) = \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{y-s_{i,y}}{x-s_{i,x}}\right) \\ \arctan\left(\frac{z-s_{i,z}}{\sqrt{(x-s_{i,x})^2+(y-s_{i,y})^2}}\right) \end{pmatrix} \quad (48)$$

where $(s_{i,x}, s_{i,y}, s_{i,z}), i = 1, 2$ denote the positions of the stationary sensors. The measurement covariance can be defined as $R_n = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_{e_n}^2$.

The real initial position of the target is $(2 \text{ km}, 8 \text{ km}, 1 \text{ km})$, and the initial velocity is $(0.15 \text{ km/s}, 0.26 \text{ km/s}, 0.0 \text{ km/s})$. The prior PDF of state x_0 is assumed to be $x_0 \sim N(\hat{x}_{0|0}, \hat{P}_{0|0})$, where $\hat{x}_{0|0} = [2.1 \text{ km } 0.12 \text{ kms}^{-1} 7.95 \text{ km } 0.23 \text{ kms}^{-1} 0.95 \text{ km } 0 \text{ kms}^{-1}]^T$, $\hat{P}_{0|0} = \text{diag}[0.144 \text{ km}^2 \ 0.02^2 \text{ km}^2\text{s}^{-2} \ 0.144 \text{ km}^2 \ 0.02^2 \text{ km}^2\text{s}^{-2} \ 0 \ 0]^T$. The standard deviation of the process and the measurement noise are set to $\sigma_{m_n} = 0.01$ and $\sigma_{e_n} = 0.001$, and these noises are zero-mean Gaussian-distributed and independent. The number of particles is set to 200.

Figure 5 shows the X RMSE, Y RMSE, Z RMSE and position RMSE of the ATUKF compared with IMMKEF, IMMRBPF and TQKF. From Figure 5, we can see that the RMSE of all algorithms abruptly

increased from 30 s to 50 s, which is mainly due to the increase of the target predicted errors caused by the target maneuvering. However, in Figure 5a,c,d, it is shown that the performance of the ATUKF algorithm has outperformed the IMMEKF, IMMRBPF and TQKF. A key reason is that the proposed algorithm can incorporate the target characteristic information and current measurement information into the prior PDF, which can effectively degrade the variance of errors caused due to the maneuvering of the target. Moreover, because the flight height of target remained unchanged, from Figure 5c, the Z RMSE of all algorithms is very close.

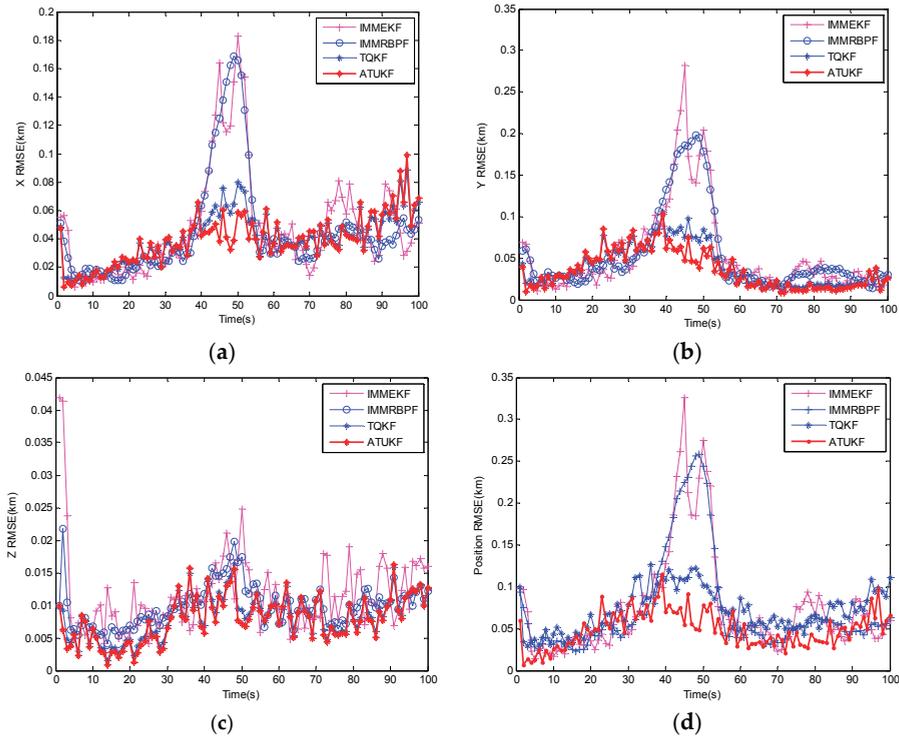


Figure 5. Root-mean-square error (RMSE) of the ATUKF, IMMEKF and IMMRBPF. (a) X RMSE; (b) Y RMSE; (c) Z RMSE; (d) position RMSE.

Finally, the computation time statistics for all algorithms are given in Table 2. In this case, all the experiments are performed by using MATLAB programming on an Intel-Core(TM)-i2-4030U processor (1.9 GHz) based on the Windows platform. In Table 2, it is shown that the computational load of the IMMRBPF is much higher than these of the IMMEKF, TQKF and ATUKF. More importantly, the ATUKF requires much less of a computation time than the TQKF. However, the computational load of the ATUKF is nearly two times higher than that of the IMMEKF.

Table 2. Computation times for all algorithms (s). BOT: Bearings-only maneuvering tracking.

Case	IMMEKF	IMMRBPF	TQKF	ATUKF
BOT	0.074	14.493	0.553	0.150

4. Conclusions

In this paper, we presented a bearings-only target tracking algorithm based on an auxiliary truncated unscented Kalman filtering (ATUKF) algorithm. Unlike the truncated unscented Kalman filtering, in the proposed algorithm, several target characteristics were introduced to construct the modified prior PDF, and the statistical linear regression was used to linearize the nonlinear non-bijective measurement function by using the sigma points. Moreover, we have developed a practical algorithm for a bearings-only target tracking system. Finally, in the simulation results, compared with the EKF, UKF, the quadrature Kalman filtering (QKF), the mixture truncated unscented Kalman filter (MTUKF) and the particle filter (PF), the ATUKF exhibits better performance. For the second case, compared with the IMM-EKF algorithm, the IMM-RBPF algorithm and the TQKF algorithm, the ATUKF algorithm not only improves the performance of the tracker, but significantly reduces the computation time.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (Grant Nos. 61301074, 61271107, 61375015), Natural Science Foundation of the Guangdong Province of China (Grant No. S2012010009417), Defense Advance Research Fund Project (Grant No. 91400***501140C80340).

Author Contributions: Liang-Qun Li conceived and designed the experiments; Xiao-Li Wang performed the experiments; Zong-Xiang Liu analyzed the simulated results; Wei-Xin Xie contributed analysis tools; Liang-Qun Li and Xiao-Li Wang wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Battistini, S.; Shima, T. Differential games missile guidance with bearings-only measurements. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 2906–2915. [CrossRef]
- Li, W.L.; Jia, Y.M. An information theoretic approach to interacting multiple model estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 1811–1825. [CrossRef]
- Zhang, Q.; Song, T.L. Improved Bearings-Only Multi-Target Tracking with GM-PHD Filtering. *Sensors* **2016**, *16*, 1469. [CrossRef] [PubMed]
- Zhao, Y.; Patwari, N. Robust estimators for variance-based device-free localization and tracking. *IEEE Trans. Mob. Comput.* **2015**, *14*, 2116–2129. [CrossRef]
- Viani, F.; Robol, F.; Polo, A.; Rocca, P.; Oliveri, G.; Massa, A. Wireless architectures for heterogeneous sensing in smart home applications-Concepts and real implementations. *Proce. IEEE* **2013**, *101*, 2381–2396. [CrossRef]
- Li, L.; Xie, W. A Novel Auxiliary Quadrature Particle Filtering Algorithm Based on Target Characteristic. *Acta Electron. Sin.* **2014**, *42*, 2069–2074. (In Chinese).
- Blom, H.A.P.; Bar-shalom, Y. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Trans. Autom. Control* **1988**, *33*, 780–783. [CrossRef]
- Syamantak, D.G.; Yu, J.Y.; Mahendra, M.; Mark, C.; Mark, M. Comparison of Angle-only Filtering Algorithms in 3D Using EKF, UKF, PF, PFF, and Ensemble KF. In Proceedings of the 18th International Conference on Information Fusion Washington, Washington, DC, USA, 6–9 July 2015; pp. 1649–1656.
- Gustafsson, F.; Hendeby, G. Some relations between extended and unscented Kalman filters. *IEEE Trans. Signal Process.* **2012**, *60*, 545–555. [CrossRef]
- Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proce. IEEE* **2004**, *92*, 401–422. [CrossRef]
- Ángel, F.G.-F.; Mark, R.M.; Jesús, G. Truncated Unscented Kalman Filtering. *IEEE Trans. Signal Process.* **2012**, *60*, 3372–3386.
- García-Fernández, A.F.; Morelande, M.R.; Grajal, J. Mixture truncated unscented Kalman filtering. In Proceedings of the 15th International Conference on Information Fusion, Singapore, 9–12 July 2012; pp. 479–486.
- Straka, O.; Duník, J.; Šimandl, M. Truncation nonlinear filters for state estimation with nonlinear inequality constraints. *Automatica* **2012**, *48*, 273–286. [CrossRef]
- Boada, B.L.; Garcia-Pozuelo, D.; Boada, M.J.L.; Diaz, V. A Constrained Dual Kalman Filter Based on pdf Truncation for Estimation of Vehicle Parameters and Road Bank Angle: Analysis and Experimental Validation. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1006–1016. [CrossRef]

15. Gordon, N.J.; Salmond, D.J.; Smith, A.F.M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceed. Radar Sonar Navig.* **1993**, *140*, 107–113. [CrossRef]
16. Sutharsan, S.; Kirubarajan, T.; Lang, T.; McDonald, M. An Optimization-Based Parallel Particle Filter for Multitarget Tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 1601–1618. [CrossRef]
17. Hlinka, O.; Hlawatsch, F.; Djuric, P.M. Distributed Particle Filtering in Agent Networks. *IEEE Signal Process. Mag.* **2013**, *30*, 61–81. [CrossRef]
18. Yang, T.; Laugesen, R.S.; Mehta, P.G.; Meyn, S.P. Multivariable feedback particle filter. *Automatica* **2016**, *71*, 10–23. [CrossRef]
19. Boers, Y.; Driessen, J.N. Interacting multiple model particle filter. *IEEE Proc. Radar Sonar Navig.* **2003**, *150*, 344–349. [CrossRef]
20. Keshavarz-Mohammadiyan, A.; Khaloozadeh, H. Adaptive IMMPPF for bearing-only maneuvering target tracking in Wireless Sensor networks. In Proceedings of the 4th International Conference on Control, Instrumentation, and Automation (ICCIA), Qazvin, Iran, 27–28 January 2016; pp. 6–11.
21. Li, L.; Xie, W.; Huang, J.; Huang, J. Multiple Model Rao-Blackwellized Particle Filter for Maneuvering Target Tracking. *Int. J. Def. Sci.* **2009**, *59*, 197–204.
22. Yu, J.Y.; Coates, M.J.; Rabbat, M.G.; Blouin, S. A Distributed Particle Filter for Bearings-Only Tracking on Spherical Surfaces. *IEEE Signal Process. Lett.* **2016**, *23*, 326–330. [CrossRef]
23. Taghavi, E.; Tharmarasa, R.; Kirubarajan, T.; McDonald, M. Multisensor-multitarget bearing-only sensor registration. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 1654–1666. [CrossRef]
24. Li, L.; Xie, W.; Liu, Z. Bearings-only maneuvering target tracking based on truncated quadrature Kalman filtering. *Int. J. Electron. Commun.* **2015**, *69*, 281–289.
25. Ienkaran, A.; Simon, H. Discrete-Time Nonlinear Filtering Algorithms Using Gauss-Hermit Quadrature. *Proc. IEEE* **2007**, *95*, 953–976.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Flexible Fusion Structure-Based Performance Optimization Learning for Multisensor Target Tracking

Quanbo Ge ^{1,2}, Zhongliang Wei ^{3,*}, Tianfa Cheng ¹, Shaodong Chen ⁴ and Xiangfeng Wang ⁵

¹ School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China; qbge@hdu.edu.cn (Q.G.); ctf811412@163.com (T.C.)

² State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

³ School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001, China

⁴ Science and Technology on Electro-Optic Control Laboratory, Luoyang Institute of Electro-Optical Equipment of Avic, Luoyang 471000, China; Chen-shao-dong@163.com

⁵ Shanghai Key Lab for Trustworthy Computing, East China Normal University, Shanghai 200062, China; xfwang@sei.ecnu.edu.cn

* Correspondence: zhlwei@aust.edu.cn; Tel.: +86-571-8691-5193

Academic Editors: Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang

Received: 27 January 2017; Accepted: 22 April 2017; Published: 6 May 2017

Abstract: Compared with the fixed fusion structure, the flexible fusion structure with mixed fusion methods has better adjustment performance for the complex air task network systems, and it can effectively help the system to achieve the goal under the given constraints. Because of the time-varying situation of the task network system induced by moving nodes and non-cooperative target, and limitations such as communication bandwidth and measurement distance, it is necessary to dynamically adjust the system fusion structure including sensors and fusion methods in a given adjustment period. Aiming at this, this paper studies the design of a flexible fusion algorithm by using an optimization learning technology. The purpose is to dynamically determine the sensors' numbers and the associated sensors to take part in the centralized and distributed fusion processes, respectively, herein termed *sensor subsets selection*. Firstly, two system performance indexes are introduced. Especially, the survivability index is presented and defined. Secondly, based on the two indexes and considering other conditions such as communication bandwidth and measurement distance, optimization models for both single target tracking and multi-target tracking are established. Correspondingly, solution steps are given for the two optimization models in detail. Simulation examples are demonstrated to validate the proposed algorithms.

Keywords: flexible fusion structure; mixed fusion method; combinatorial optimization; sensor subsets selection; tracking accuracy; system survivability

1. Introduction

The rapid development of some key technologies—for example, communication technology, sensor technology, data processing, and so on—have promoted the research into applications of wireless sensor network systems. According to different demands, there are mainly two types including passive and active networks. Due to their different working principles, the data processing methods are clearly different. Therefore, we should have different views when designing data fusion algorithms for sensor networks. Actually, it is necessary that fusion algorithms should have adaptive function because of the complex application background. Namely, the fusion structure of the task network

system should be flexible. This means that the task network system can dynamically determine and choose the sensors taking part in the fusion processes. Namely, the flexible fusion also indicates that sensor numbers are time-varying for centralized and distributed fusion methods. Then, most of the traditional fusion methods cannot be directly used to deal with dynamical sensor selection. To some extent, the sensor selection belongs to the sensor management domain. It is important and significative to study flexible fusion methods for wireless task network systems.

Multisensor fusion is a fundamental technique for networked information systems (NIS), which can fuse different kinds of measurement data from multiple sensors. It has the advantage of reducing the uncertainty of target perception and improving the performance of the NIS [1–6]. For instance, designing multi-sensor data fusion algorithms in order to improve target tracking learning system performance has recently been one of the popular topics in the NIS area. The traditional fusion technologies mainly include centralized and distributed fusion methods. The two approaches have different performance and application background. Accordingly, it is a good way to combine the two fusion approaches—the mixed fusion structure with the centralized and distributed fusions is more effective for task network systems. Thereby, the issue of how to ensure that the sensors to respectively take part in the centralized fusion and distributed fusion is important and challenging. In addition, the adopted basic nonlinear filter also affects the fusion performance for nonlinear systems. An improved self-adaptive unscented Kalman algorithm was presented in [7] to ameliorate the stability of target tracking. A particle filter algorithm is proposed based on optimizing the dynamic neighborhood self-adaptive particle in order to raise target tracking accuracy in [8]. Compared with the traditional Kalman filter, a fixed gain Kalman filter which was studied in [9] had better performance from the perspective of computational cost. Further, based on the self-adaptive neuro-fuzzy inference system, In [10], Ma et al. presented an improved Kalman filtering algorithm to reduce the tracking error. Sun et al. [11] presented an improved extended Kalman filter-based target tracking learning algorithm. However, overall research on the sensor fusion structure is still lacking; until recently, most works were mainly based on single and fixed fusion methods. They have not considered the case that an initial sensor fusion structure would not be adaptive to the whole tracking process because of the richness of the target tracking system, while simultaneously lacking a sensor fusion structure recombination design. The focus of this paper is mainly to design and solve an optimization model on sensor subsets selection, so only the traditional unscented Kalman filter (UKF) is used. In fact, the nonlinear filters mentioned above can be used in our optimization solution to improve associated fusion estimation and tracking performance.

In this paper, the flexible fusion structure concept is introduced to improve the universality of the learning system. Based on our early work in [12], we will explain the concept, formation, and application scenes of the flexible fusion structure in detail, and further analyze the advantages of flexible fusion structure relative to the fixed fusion structure. Here, a fixed fusion structure means that once the fusion method (e.g., centralized fusion or distributed fusion) is determined, neither the fusion method nor the sensors taking part in the fusion process are changed during the whole working time. Clearly, this approach does not satisfy the practical requirements of engineering applications. However, because the flexible fusion structure has a dynamical adjustment function, it has better self-adaptive adjustment ability than the fixed structure. Additionally, it can quickly and flexibly regulate the system resources allocation to respond to the change of the tracking situation. Aiming at the above conditions and based on the earlier work [12], we focus on this popular research topic (i.e., flexible fusion structure for target tracking or state estimation). In order to achieve self-adaptive adjustment ability, some available models and algorithms are analyzed in this manuscript, and the main contributions are as follows:

- Two indexes (tracking accuracy and survivability) are introduced to integrally describe system performance in Section 2.

The current work gives substantial attention to the tracking accuracy. However, the survivability index is seldom discussed. In this work, a definition of survivability is presented and a detailed computation method is also given.

- The optimization models are established for sensor subsets selection for single target and multi-target tracking in Section 3.1. Based on the two performance indexes, two optimization models with multiple constraints are creatively designed. Clearly, the optimization model based on single target tracking is the foundation of multi-target tracking.
- The solutions of the optimization models are also given and the detailed solution steps are clearly given in Section 3.2.

The rest of the paper is organized as follows. The problem formulation—including the introduction of the task network system, system description, and fusion methods—is given in Section 1. Section 2 introduces the two performance indexes: the tracking accuracy and the survivability. In Section 3, the dynamic sensor subsets optimization selection problem—including optimization models and solutions—is studied under the flexible fusion structure. Simulation examples are demonstrated in Section 4. Finally, we conclude the paper in Section 5.

2. Problem Formulation

2.1. Task Network System

A networked information system (NIS) connects all the information units within a given domain in order to construct a real-time and high-speed information system. Information fusion is one of the important techniques used to establish this kind of NIS, and the basic networked information fusion structure is shown in Figure 1. The information fusion center communicates with the local sensor nodes by data link, and then the system fusion center sends commands to each sensor node and receives the measurement data or the local fusion estimation data from sensor nodes. Further global data fusion will be constructed. There are two limitations to this structure:

- Each sensor node can only track a limited number of targets;
- The fusion center can only process a limited amount of sensor measurement data with respect to the limited communication bandwidth and the computing capacity.

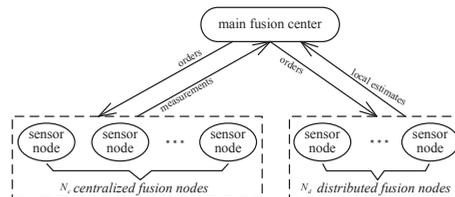


Figure 1. The basic information fusion structure.

The target tracking process is not stable, while the tracking situation is complex, so the system performance indexes are affected by the internal and external unknown factors. In order to achieve an optimal or suboptimal affected system state, we need to regulate the configuration of system surplus resources and change the fusion method of sensor nodes. In the following, we will analyze the target tracking fusion system in detail and design a new system optimization formulation.

2.2. System Description and Fusion Methods

The target tracking system includes M warplanes (M dynamic sensor nodes), and each warplane is equipped with L same groups of sensors. $B(kbps)$ denotes the total system communication bandwidth resource. In order to reduce signal caused risk, the system adopts the passive tracking method, with the data receiving period being T .

First, we briefly give the nonlinear target tracking system state model as follows:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{w}_k, \tag{1}$$

$$\mathbf{z}_{k,i} = \mathbf{h}_i(\mathbf{x}_k) + \mathbf{v}_{k,i}, \quad i = 1, 2, \dots, ML, \tag{2}$$

where (1) and (2) are tracking state equation and measurement equation; \mathbf{x}_k is state vector and $\mathbf{z}_{k,i}$ is measurement vector; $\mathbf{f}(\mathbf{x}_k)$ and $\mathbf{h}_i(\mathbf{x}_k)$ are state transition function matrix and state measurement function matrix, respectively; process noise \mathbf{w}_k and measurement noise $\mathbf{v}_{k,i}$ are zero mean white Gaussian noise with the covariance \mathbf{Q}_k and $\mathbf{R}_{k,i}$, respectively, while the measurement noises between each sensor are uncorrelated.

In this work, the UKF is taken as the basic data filtering method, and the details refer to [13–15].

Suppose in time period t that there are $m_{c,t}$ and $m_{d,t}$ number of centralized fusion nodes and distributed fusion nodes, respectively, and $m_{c,t} + m_{d,t} \leq M$ (note: subscript notation “ c ” denotes the part of centralized fusion and “ d ” denotes the part of distributed fusion). The active sensor numbers are $n_{i_c,t}$ ($i_c = 1, 2, \dots, m_{c,t}$) and $n_{i_d,t}$ ($i_d = 1, 2, \dots, m_{d,t}$), and

$$\begin{cases} n_{c,t} = \sum_{i_c=1}^{m_{c,t}} n_{i_c,t} \\ n_{d,t} = \sum_{i_d=1}^{m_{d,t}} n_{i_d,t} \end{cases}, \quad \text{and} \quad n_{c,t} + n_{d,t} \leq M \cdot L. \tag{3}$$

The adopted dimension expansion fusion method and local estimate weighted fusion method for system centralized fusion nodes and distributed fusion nodes are as follows [16]:

- Dimension Expansion Fusion Method

Integrate N_c measurement equations into a large measurement equation

$$\mathbf{Z}_k = \mathbf{H}(\mathbf{x}_k) + \mathbf{V}_k, \tag{4}$$

and

$$E\{\mathbf{V}_k\} = 0, E\{\mathbf{V}_k \mathbf{V}_k^T\} = \mathbf{R}_k^*,$$

where

$$\begin{aligned} \mathbf{Z}_k &= \begin{bmatrix} \mathbf{z}_{k,1}^T & \mathbf{z}_{k,2}^T & \cdots & \mathbf{z}_{k,n_{c,t}}^T \end{bmatrix}^T, \\ \mathbf{H}_k &= \begin{bmatrix} \mathbf{h}_1^T(\mathbf{x}_k) & \mathbf{h}_2^T(\mathbf{x}_k) & \cdots & \mathbf{h}_{n_{c,t}}^T(\mathbf{x}_k) \end{bmatrix}^T, \\ \mathbf{V}_k &= \begin{bmatrix} \mathbf{v}_{k,1}^T & \mathbf{v}_{k,2}^T & \cdots & \mathbf{v}_{k,n_{c,t}}^T \end{bmatrix}^T, \\ \mathbf{R}_k^* &= \begin{bmatrix} \mathbf{R}_{k,1} & 0 & \cdots & 0 \\ 0 & \mathbf{R}_{k,2} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \mathbf{R}_{k,n_{c,t}} \end{bmatrix}. \end{aligned}$$

Based on the state equation and the measurement equation, applying the basic UKF algorithm, the multi-sensor centralized dimension expansion fusion estimator is

$$\begin{cases} \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{Z}_k - \hat{\mathbf{Z}}_{k|k-1}), \\ \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{zz,k|k-1} \mathbf{K}_k^T, \end{cases} \tag{5}$$

where

$$\begin{cases} \mathbf{P}_{xz,k|k-1} = \sum_{j=0}^{2n} \omega_{cov,j} \left(\mathbf{O}_{k|k-1,j} - \hat{\mathbf{x}}_{k|k-1} \right) \\ \quad \times \left(\mathbf{Y}_{k|k-1,j} - \hat{\mathbf{Z}}_{k|k-1} \right)^T, \\ \mathbf{P}_{zz,k|k-1} = \sum_{j=0}^{2n} \omega_{cov,j} \left(\mathbf{Y}_{k|k-1,j} - \hat{\mathbf{Z}}_{k|k-1} \right) \\ \quad \times \left(\mathbf{Y}_{k|k-1,j} - \hat{\mathbf{Z}}_{k|k-1} \right)^T + \mathbf{R}_k^* \\ \mathbf{K}_k = \mathbf{P}_{xz,k|k-1} \mathbf{P}_{zz,k|k-1}^{-1} \end{cases} \quad (6)$$

- Local Estimate Weighted Fusion Method

The local estimate weighted fusion estimator is

$$\begin{cases} \hat{\mathbf{x}}_{k|k} = \sum_{i_d=1}^{m_{d,t}} \frac{\mathbf{P}_{k|i_d}^{-1}}{\mathbf{P}_{k|k}^{-1}} \hat{\mathbf{x}}_{k|i_d}, \\ \mathbf{P}_{k|k}^{-1} = \sum_{i_d=1}^{m_{d,t}} \mathbf{P}_{k|i_d}^{-1}. \end{cases} \quad (7)$$

From (3) to (6), we can get the state estimate $\hat{\mathbf{x}}_{k|k,c}$ and the estimate error covariance $\mathbf{P}_{k|k,c}$ of the centralized fusion part; from (7), we can get the state estimate $\hat{\mathbf{x}}_{k|k,d}$ and the estimate error covariance $\mathbf{P}_{k|k,d}$ of the distributed fusion part, so the global fusion results of the system are

$$\hat{\mathbf{x}}_{k|k,sys} = \frac{\mathbf{P}_{k|k,c}^{-1}}{\mathbf{P}_{k|k,sys}^{-1}} \hat{\mathbf{x}}_{k|k,c} + \frac{\mathbf{P}_{k|k,d}^{-1}}{\mathbf{P}_{k|k,sys}^{-1}} \hat{\mathbf{x}}_{k|k,d} \quad (8)$$

$$\mathbf{P}_{k|k,sys}^{-1} = \mathbf{P}_{k|k,c}^{-1} + \mathbf{P}_{k|k,d}^{-1} \quad (9)$$

As we know, there are several ways to exchange data in task network systems. Because this paper considers a kind of special air task network system which is strictly limited to communication among nodes, it only considers a simple approach. Namely, all available sensors send local information to the fusion center; for example, the nodes under the centralized fusion mode send the measurements, and the nodes under the distributed fusion mode send the local estimates. There is no commutation among local available sensors. All fusion operations are done in the fusion center, regardless of the centralized fusion and the distributed fusion. Thereby, the centralized fusion and the distributed fusion are carried out in parallel with the data communication; for example, the group method with multiple CPUs can be used. However, for the fusion process with the centralized and the distributed information in the fusion center, there are several ways to integrate the information. Strictly speaking, the fusion process is not in parallel with the fusion center for the centralized fusion and the distributed fusion. Commonly, the fusion is performed under a given sequential rule, and it is highly effective because the CPU in the fusion center has strong computation ability.

3. Analysis of System Performance Indexes

The purpose of this paper is to improve the self-adjustment ability of the target tracking system, and the system optimization goal is to maximize the system performance within certain resources to solve the optimal configuration problem of system resources. Target tracking accuracy and system survivability are the two most important performance judgements of the target tacking fusion system [17]. The system performances have a close relationship with system resources allocation, and the above two indexes will be analyzed qualitatively and quantitatively in detail as follows.

3.1. Tracking Accuracy

Accurate target location is the primary task of a target tracking system, which reflects on the tracking accuracy index that to a certain extent determines the system's overall performance. The tracking accuracy is related to sensor performance, measurement data volume and quality, fusion algorithm and fusion structure, and other external uncontrollable factors. External factors cannot be artificially controlled, so, in order to guarantee the system performance under changing external conditions, we should regulate the deployment and allocation of internal system resources and modify the data fusion method of each sensor node self-adaptively.

Multi-sensor technology improves the system performance to a large degree, but the system becomes more complex. In target tracking, it is necessary to solve the problem of optimal sensor subsets selection. More uncertain sensors lead to better results, and the best way is to choose the optimal combinations of sensors and the optimal combination of fusion methods for each target in order to obtain optimal tracking performance [18–20]. Therefore, dynamic sensor management is one of the important links for a sensor network system, while controlling the sensors at optimal working status can greatly improve the system performance.

Compared with distributed fusion, centralized fusion has better fusion accuracy, and in this paper we use the hybrid fusion method to process data based on the UKF. The hybrid fusion method combines the advantages of the centralized and distributed fusion methods, and it is a supplement of those two fusion methods. The system data fusion method is based on the UKF, so the convergence expectation of fusion estimate error covariance can be obtained as the measure standard of target tracking accuracy [21–23]. When M sensors all adopt the centralized fusion method, the system has the best tracking accuracy; meanwhile, when the system has only one sensor node working, the tracking accuracy is worst.

Suppose the system fusion estimation error covariance is $\mathbf{P}_{e,t}$ as calculated by (9). The upper and lower limits of $\text{tr}(\mathbf{P}_{e,t})$ are tr_a and tr_b , respectively; i.e.,

$$tr_a < \text{tr}(\mathbf{P}_{e,t}) < tr_b,$$

where

$$\begin{cases} tr_a \geq \text{tr}(\mathbf{P}_{min}), \\ tr_b \leq \text{tr}(\mathbf{P}_{max}), \end{cases}$$

where $\text{tr}(\mathbf{P})$ denotes the trace of matrix \mathbf{P} ; $\text{tr}(\mathbf{P}_{min})$ is the convergence expectation of fusion estimation error covariance, while all the sensor nodes adopt the centralized fusion method; $\text{tr}(\mathbf{P}_{max})$ is the convergence expectation of fusion estimation error covariance, while the system has only one sensor node working.

3.2. System Survivability

The “survivability” index is used to express the possibility that the task system cannot be discovered by the non-cooperative target. The survivability can be used for the nodes and the system. In this paper, it is used for the task network system. This is because the whole network system should be discovered with a large probability once one of the nodes has been found by the non-cooperative target [24]. The factors influencing survivability index include data communication between the network nodes/platforms and the fusion center, radar radiation, etc. For simplicity and considering the passive tracking, we only consider the influence of data communication traffic on the survivability index. For this, the “survivability” is in relation with fusion architecture design. The communication between the fusion center and the nodes with only a passive tracking function is a main event leading to being found for the task network. Thereby, we use the communication time to formulate the survivability of the air task network. Intuitively, longer communication time between the center and the nodes or among nodes means worse survivability for the task network system—namely, they have

an inverse relation. In other words, because there is more communication between the fusion center and the local nodes, the fusion center can obtain more available information on the non-cooperative target. Obviously, the fusion accuracy can be improved due to more information. Likewise, more information communication will lead to a greater probability of being detected by non-cooperative targets, namely the risk of being discovered by non-cooperative targets should be increased and the survivability of the tracking system should be reduced. In contrast, less communication means that the information taken by the fusion center is less and the fusion accuracy should be decreased. At the same time, the survivability should be increased because of less communication and the probability of it being detected and discovered should be reduced.

For a given fusion period, the local sensors have several possible samples/measurements. For the centralized fusion nodes, there are many data transmission operations from local sensors to the fusion center in a given fusion period. For the distributed fusion nodes, there is only one transmission operation in a period because many samples can be processed by a local processor to form a unified local estimate, which can be transmitted to the fusion center. If the centralized fusion node is allocated too much in the system structure design, it will bring out too much data transmission traffic, which will have a great influence on system survivability, and our planes will be easily exposed to the nonoperative targets; therefore, we should reduce the number of centralized fusion nodes. In order to guarantee the requirement of tracking accuracy, we have to increase the number of centralized fusion nodes, which leads to a mutual restriction relation between the system survivability index and the tracking accuracy index. Consequently, we should adjust the allocation of the node fusion method under the given conditions in the context of the actual conditions.

In this paper, we consider the survivability index to be mainly determined by the data communication traffic, which can be measured by the data communication times c_t between local sensor nodes and the fusion center. Here, we do not consider the fully decentralized fusion structure and there is no communication among local sensors. Based on the experience, the survivability index changes little within the limited extent of data transmission times, and with the rapid increase of data transmission times, the survivability index declines quickly. Due to the inverse relation mentioned above, we considered several kinds of decreasing functions. Through graphical simulation analysis, the amplitude–frequency characteristics function of the first order inertia link is comparatively appropriate if it could be properly improved. Accordingly, in terms of background knowledge and experience, two modifications have been done to obtain an available survivability index. The first is to modify the quadratic as a cube, keeping the root sign the same, and the second is to adopt a logarithmic operation to realize dimensionless and standardization effects. The logarithmic form could be considered to be derived from the logarithmic amplitude–frequency characteristics function.

Then, according to the explanation mentioned above, we can get the time computation formula of one communication operation from transmission to reception, which can be expressed by a third-order inertia logarithmic function

$$s_t = \lg \frac{10}{\sqrt[3]{1 + \lambda^3 c_t^3}}, 0 < s_t \leq 1, \quad (10)$$

where $0 < \lambda \leq 1$ is the function attenuation coefficient that determines the function attenuation trend—it is a positive decimal and the λ value is different in different systems. Certainly, other methods for the design of the survivability index may exist, and a comparison study is very important and significant in future work.

In order to realize the sensor management function or flexible fusion structure (namely, to determine the sensor subsets under the centralized and distributed fusion frames), it is important to construct an index to describe the communication time of the whole task network system in a fusion period. This index is taken as the base to optimally solve the sensor subsets. From a normal viewpoint, greater communication time means a greater probability of being discovered by the noncooperative target. Actually, there are many possible ways to construct the system commutation time. Here, we simply take the summation of commination times of all used nodes in a fusion period as the system

commutation time. In other words, the system commutation time is composed of two parts, which are: the system centralized fusion part $c_{c,t} = n_{c,t}$ and the system distributed fusion part $c_{d,t} = m_{d,t}$. Then, the system communication time is expressed by

$$c_{sys,t} = c_{c,t} + c_{d,t} = n_{c,t} + m_{d,t}.$$

According to different values of λ , we plot the change curve of s_t as Figure 2. As shown in the figure, the change curve of s_t basically conforms to the qualitative analysis change requirements of the survivability index, which indicates that the design function of s_t is feasible. It should be noted that although there are many ways to design the system communication time index, it is not naturally influenced to establish and solve the optimization model.

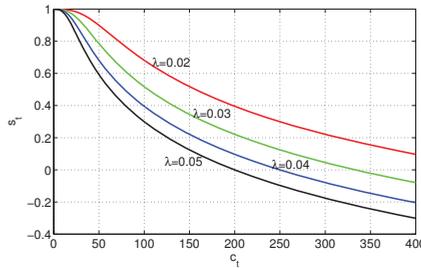


Figure 2. The change curve of s_t .

It is necessary to consider the security risk factor condition for a single tracking plane, which is an important part of the system survivability index. When the threat of one enemy plane to one of our planes being greater than the safe threshold value, our tracking plane can adopt the action of switching to standby work mode to stop all the sensor activities, and use the avoiding protection method to stay at the tracking formation. Until the enemy threat becomes relatively small, the dormant tracking plane can restart work on the tracking task.

A single-platform security risk coefficient can be treated as the threat value of enemy planes to our planes; a greater threat value means greater mission risk. The method of evaluating threat value was drawn from [25]. Suppose the requirement of system initial risk coefficient safe threshold value is sr_0 , and the real-time evaluation of security risk coefficient is $sr_{i,t}$, ($i = 1, 2, \dots, M$), and

$$sr_{i,t} < sr_0, \quad i = 1, 2, \dots, M.$$

4. Dynamic Sensor Subsets Selection Under Flexible Fusion Structure

4.1. Establishment of Optimization Model for Sensor Subsets Selection

Because of the time-varying situation of the task network system induced by moving nodes and non-cooperative target, and limitations such as communication bandwidth and measurement distance, it is necessary to dynamically adjust the system fusion structure, including sensors and fusion methods taking part in the fusion for a given adjustment period. The target tracking accuracy is measured by the estimated error covariance, so the tracking accuracy of this paper is a low-quality index, and it can be standardized by the index quantitative method of cost, defined by the standardized function $P_p(\text{tr}(\mathbf{P}_{e,t}))$ as the target tracking accuracy performance value function, and

$$P_p(\text{tr}(\mathbf{P}_{e,t})) = \begin{cases} 1, & \text{tr}(\mathbf{P}_{e,t}) \leq \text{tr}(\mathbf{P}_{min}), \\ \frac{\text{tr}(\mathbf{P}_{max}) - \text{tr}(\mathbf{P}_{e,t})}{\text{tr}(\mathbf{P}_{max}) - \text{tr}(\mathbf{P}_{min})}, & \text{tr}(\mathbf{P}_{min}) < \text{tr}(\mathbf{P}_{e,t}) < \text{tr}(\mathbf{P}_{max}), \\ 0, & \text{tr}(\mathbf{P}_{e,t}) \geq \text{tr}(\mathbf{P}_{max}). \end{cases} \quad (11)$$

The above formula is a piecewise function which expresses three cases on $\mathbf{P}_{e,t}$, \mathbf{P}_{min} , and \mathbf{P}_{max} . A larger value of $P_p(\text{tr}(\mathbf{P}_{e,t}))$ means the system has better tracking accuracy. The goal of the target tracking system dynamic sensor nodes management is to choose the best combination of the sensor nodes fusion method within the performance index requirement extent, and it has to meet the necessary constraints, which brings the maximum performance into the target tracking system. In summary, we can design a kind of objective optimization model of sensor nodes' dynamic management:

$$f(m_{c,t}, m_{d,t}) = \arg \max_f (P_p(\text{tr}(\mathbf{P}_{e,t}))), \quad (12)$$

$$s.t. \begin{cases} 0 < m_{c,t} + m_{d,t} \leq M, \\ tr_a < \text{tr}(\mathbf{P}_{e,t}) < tr_b, s_{min} < s_t \leq 1, \\ sr_{i,t} < sr_0, r_{i,t} < D_i, \\ r_{bw,t} \leq 1, \end{cases}$$

where $r_{i,t} < D_i$ is the constraint of sensor measurement distance, D_i is the maximum measurement distance of each sensor node; $r_{bw,t} \leq 1$ is the constraint of data communication bandwidth, $r_{bw,t}$ is the consumption proportion of bandwidth resource,

$$r_{bw,t} = \frac{n_{c,t} \times b + m_{d,t} \times b}{B},$$

where b is the occupied communication bandwidth size of one sensor to transmit the measurement data in one time period. The network adjustment principle is shown by optimization formulation Equation (12). The solution of optimization Equation (12) is the numbers and the associated sensors that take part in the centralized and distributed fusion processes.

4.2. Multi-Step Solution of Multi-Constraint Optimization Model

The multi-airborne sensor nodes allocation is a problem of multi-target NP combination optimization. For the massive case, with the increase of targets and sensor nodes, it is difficult to solve the model directly, which will cause the problem of "combination explosion", and it needs a large amount of computation time and storage space, and, given this, it is even possible that the model will remain unsolved. So we adopt the step-by-step solution strategy to gradually reduce the solution space based on the model constraints, and the optimal solution can be obtained.

For the single target system, the solution steps are as follows:

- (1) Based on the constraint of sensor node measurement distance, mark off the distant available sensor node subset \mathbf{S}_1 ;
- (2) Check whether the subset \mathbf{S}_1 is consistent with the constraint of single plane security risk to get the security risk available subset \mathbf{S}_2 ;
- (3) Solve all the possible groups of $m_{c,t}$ and $m_{d,t}$ under the constraints of $\text{tr}(\mathbf{P})$, s , and r_{bw} . If $m_{c,t} + m_{d,t} > \text{size}(\mathbf{S}_2)$, there is no optimization solution, and if the situation is allowable, we can turn back to step (1) or step (2) to widen the constraint extent and proceed to solve the next step; when the $m_{c,t} + m_{d,t} \leq \text{size}(\mathbf{S}_2)$, the model has solutions, and to get the optimal solution of $m_{c,t}$ and $m_{d,t}$ through the objective function $f(m_{c,t}, m_{d,t})$;
- (4) Allocation of $m_{c,t}$ and $m_{d,t}$ in subset \mathbf{S}_2 : firstly to allocate the $m_{c,t}$, the principle of which is to select the sensor nodes that are closer to the fusion center; if there are two sensor nodes whose distances are equidistant, choose the node that has the litter security risk coefficient; then, it is the turn of $m_{d,t}$. Its principle is the same as with $m_{c,t}$, but the allocation range is the remaining sensor nodes of subset \mathbf{S}_2 .

For a multi-target system, before the above steps, we need to allocate the optimal sensor node subsets for every target. The allocated sensor nodes of each target should not be more than the average of the total number of nodes for all targets, and every node should be allocated to a target.

The allocation principle is to maximize the threat values of airborne planes relative to targets, and the objective optimization model is

$$\delta^*(t) = \arg \min_{\delta} \sum_{j=1}^N \sum_{i=1}^M \delta_{ij,t} r_{ij,t}, \tag{13}$$

$$s.t. \begin{cases} \delta_{ij,t} \in \{0, 1\}, \sum_{j=1}^N \delta_{ij,t} = 1, \\ \lfloor \frac{M}{N} \rfloor \leq \sum_{i=1}^M \delta_{ij,t} \leq \lfloor \frac{M}{N} \rfloor + 1, \end{cases}$$

where N is the number of targets and $\delta_{ij,t}$ is the allocation matrix of sensor node i to target j . $\delta_{ij,t} = 1$ denotes that the sensor node i is tracking to the target j , otherwise the sensor node i is not allocated. $r_{ij,t}$ is the relative distance of the sensor node i to the target j . So, for the multi-target situation, the objective optimization model of sensor nodes dynamic management is updated as

$$f^*(m_{c,t}^j, m_{d,t}^j) = \arg \max_f P_f \left(\text{tr} \left(\sum_{j=1}^N (\mathbf{P}_{e,t}^j)^{-1} \right)^{-1} \right), \tag{14}$$

$$s.t. \begin{cases} 0 < m_{c,t}^j + m_{d,t}^j \leq \lfloor \frac{M}{N} \rfloor + 1, \\ tr_a^j < \text{tr} \mathbf{P}_{e,t}^j < tr_b^j, s_{min} < s_t \leq 1, \\ sr_{ij,t} < sr_0, r_{ij,t} < D_i, \\ r_{bw,t} \leq 1. \end{cases}$$

The $\delta_{ij,t}$ in the multi-target model that can be obtained by using the ant colony optimization algorithm (ACO) [26,27]. The ACO is a metaheuristics bionic optimization algorithm that has strong applicability in terms of solving discrete combinatorial problems. In the solution problem of $\delta_{ij,t}$ in this paper, the $\max_{\delta} \sum_{j=1}^N \sum_{i=1}^M \delta_{ij,t} r_{ij,t}$ can be treated as the elicitation function ϕ_{ij} , based on the $\sum_{j=1}^N \delta_{ij,t} = 1$ to setting the tabu table $T_{yes,j}$ of ants' search targets. The ants firstly randomly generate the target searching sequence, then quickly obtain the sensor node allocation subsets of every target, and finally determine the optimal search path through multi-iteration, and the near-optimal solution of the objective function is obtained.

5. Simulation

In order to verify the feasibility of the designed system optimization model, in this section, we demonstrate the simulation in two different situations: one is the single target tracking situation, the other is the multi-target tracking situation. It analyzes the simulation calculation results in these two situations. This paper considers two indexes (tracking accuracy and survivability) to express the system tracking performance. Commonly, this kind of study only uses the tracking accuracy index. Thereby, our scene covers most of the current studies. However, it does not compare the case with only the tracking accuracy index or the case with two indexes in the simulation section.

5.1. Single Target Tracking Situation

In time period t , our command center sends $M = 6$ reconnaissance planes to track an enemy plane T_1 that is in our airspace. Assuming the non-cooperative plane has an approximately uniform motion on the X -axis and it has an approximately uniform motion on the Y -axis as well, Figure 3 is the radar map of the enemy's and our initial states.

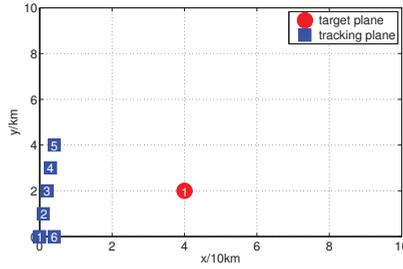


Figure 3. Radar map of enemy and friend initial states.

The target state is $\mathbf{x} = [x, y, \dot{x}, \dot{y}]^T$, its initial state is $\mathbf{x}_0 = [50,000, 2000, 380, 120]^T$, and $\mathbf{P}_0 = \text{diag}(64, 10, 4, 4)$; the state transfer matrix is

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The coordinate data of our planes is shown in Figure 3, and the 2th tracking plane is the fusion center. Each plane has been allocated $L = 3$ groups of the same measurement sensors to measure the distance r_k and angle φ_k of the target. In the actual sensor measurement, there will be additive measurement noise $\mathbf{v}_{k,i}$, so in the two-dimensional radar model, the target measurement equation is

$$\begin{aligned} \mathbf{z}_{k,i} &= \mathbf{h}_i(\mathbf{x}_k) + \mathbf{v}_k = \begin{bmatrix} r_{k,i} + v_{r,k,i} \\ \varphi_{k,i} + v_{\varphi,k,i} \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{(x_k - x_{s,i})^2 + (y_k - y_{s,i})^2} + v_{r,k,i} \\ \tan^{-1} \frac{|y_k - y_{s,i}|}{|x_k - x_{s,i}|} + v_{\varphi,k,i} \end{bmatrix}, i \in M \end{aligned}$$

where (x_k, y_k) is target coordinate and $(x_{s,i}, y_{s,i})$ is the i -th plane node coordinate.

Assume the system noise $\mathbf{Q}_k = \text{diag}(1, 1, 0.1^2, 0.1^2)$, the measurement noise $\mathbf{r}_{k,i} = \text{diag}(10^2, 0.1^2)$, and the measurement period $T = 0.5$. The target data fusion is done with the entirely centralized method and the single node fusion method, respectively, to get the estimated error covariance trace curve of the target state, as shown in Figure 4. So, we can estimate that the upper and lower limits of system fusion tracking accuracy are $E[\text{tr}(\mathbf{P}_{min})] \approx 2.0211$ and $E[\text{tr}(\mathbf{P}_{max})] \approx 16.2603$.

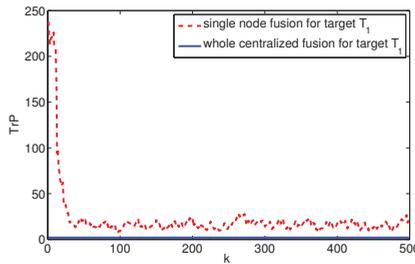


Figure 4. Trace of fusion error covariance of target T_1 .

In time period t , the performance index requirements and constraint conditions of the system target tracking task are given as follows

$$\begin{cases} 2.5000 < \text{tr}\mathbf{P}_{e,t} < 5.000, \\ 0.95 < s_t \leq 1, \\ sr_{i,t} < 0.50, r_{i,t} < 42000, \end{cases}$$

By calculating, we can know that all the airborne sensor nodes satisfy the measurement distance constraint and the security risk constraint; however, because of a mechanical failure, plane 3th is out of service. So, $\mathbf{S}_2 = \{s_i\} (i \neq 3)$ where s_i is the i -th sensor node.

Based on solution step (3), to circularly verify all the satisfied combinations of $m_{c,t}$ and $m_{d,t}$, and according to the $f(m_{c,t}, m_{d,t})$, we can select the optimal result ($m_{c,t} = 2, m_{d,t} = 3$). Based on the step (4), we can determine that the optimal allocation options are that the first and second airborne sensor nodes choose the centralized fusion method, and the fourth, fifth, and sixth nodes choose the distributed fusion method. In this allocation option, the trace of the state estimate error covariance of target T_1 is shown in Figure 5, and we can estimate $E[\text{tr}(\mathbf{P}_{e,t})] \approx 2.6026$, the system survivability index $s_t = 0.9541 (\lambda = 0.08, c_t = 9)$, and the value of optimization objective function

$$f_{\text{optimal max}}(m_{c,t} = 2, m_{d,t} = 3) = 0.9592.$$

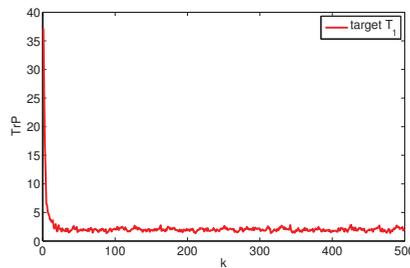


Figure 5. Optimal allocation trace of fusion error covariance of target T_1 .

5.2. Multi-Target Tracking Situation

In time period t , our command center sends $M = 9$ reconnaissance planes to track the two enemy planes T_2 and T_3 in our airspace. Figure 6 is the radar map of the enemy's and our instantaneous states. Assume that the non-cooperative planes have approximately uniform accelerated motion on the X-axis and have approximately uniform accelerated motion on Y-axis as well. The target state vector is $\mathbf{x} = [x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}]^T$, and assuming the target states are $\mathbf{x}_0^2 = [79000, 2500, 100, 25, 2, -2]^T$ and $\mathbf{x}_0^3 = [80000, 2000, 100, 25, 2, -2]^T$, and $\mathbf{P}_0^{2,3} = \text{diag}(100, 10, 1, 1, 0.1, 0.1)$, the state transfer matrix is

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

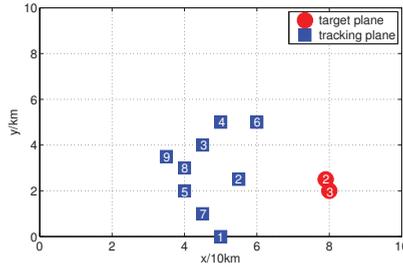


Figure 6. Radar map of enemy and friend instantaneous states.

The coordinate data of our planes is shown in Figure 6, and plane 9 is the fusion center. Each tracking plane has been allocated $L = 3$ groups of the same measurement sensors to measure the distance r_k^j and angle φ_k^j of targets. In the actual sensor measurement, there will be additive measurement noise \mathbf{v}_k^j ; so, in the two-dimensional radar model, the target measurement formula is

$$\mathbf{z}_{k,i}^j = \mathbf{h}_i(\mathbf{x}_k^j) + \mathbf{v}_{k,i}^j = \begin{bmatrix} r_{k,i}^j + v_{r,k,i}^j \\ \varphi_{k,i}^j + v_{\varphi,k,i}^j \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{(x_k^j - x_{s,i})^2 + (y_k^j - y_{s,i})^2} + v_{r,k,i}^j \\ \tan^{-1} \frac{y_k^j - y_{s,i}}{x_k^j - x_{s,i}} + v_{\varphi,k,i}^j \end{bmatrix}, \quad \begin{matrix} i \in M, \\ j \in N, \end{matrix}$$

where (x_k^j, y_k^j) is j th target coordinate, $(x_{s,i}, y_{s,i})$ is the i th plane node coordinate. Assume the system noise $\mathbf{Q}_k = \text{diag}(1, 1, 0.1^2, 0.1^2, 0.01^2, 0.01^2)$, the measurement noise $\mathbf{r}_{k,i} = \text{diag}(5^2, 0.1^2)$, and the measurement period $T = 0.5$. Undertaking the target data fusion with an entirely centralized method and single node fusion method for target T_2 and T_3 , respectively, we get the estimated error covariance trace curve of the target state as shown in Figure 7. So, we can estimate that the upper and lower limits of system fusion tracking accuracy are $E[\text{tr}(\mathbf{P}_{min})] \approx 2.0207$ and $E[\text{tr}(\mathbf{P}_{max})] \approx 40.0919$.

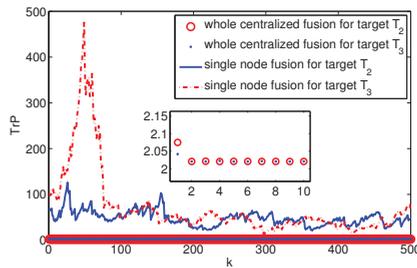


Figure 7. Trace of fusion error covariance of target T_2 and T_3 .

Based on the above simulation scene, we first allocate δ_{ij} for T_2 and T_3 by using ant colony optimization algorithms, and obtain the optimal security coefficient allocation results $(\delta_{32}, \delta_{42}, \delta_{62}, \delta_{82}, \delta_{92})$ and $(\delta_{13}, \delta_{23}, \delta_{53}, \delta_{73})$. Assume that the performance index requirements and constraint conditions of system target tracking task in time period t are

$$\begin{cases} 5.0000 < \text{tr}\mathbf{P}_{e,t}^2 < 7.0000, 6.0000 < \text{tr}\mathbf{P}_{e,t}^3 < 10.0000, \\ 0.85 < s_t \leq 1, sr_{ij,t} < 0.50, r_{ij,t} < 43000. \end{cases}$$

Based on solution step (1), we can get

$$\begin{cases} \mathbf{S}_1^2 = \{\delta_{32}, \delta_{42}, \delta_{62}, \delta_{82}\}, \\ \mathbf{S}_1^3 = \{\delta_{13}, \delta_{23}, \delta_{53}, \delta_{73}\}. \end{cases}$$

Because $sr_{16,t} = 0.546 > 0.50$, the sixth node does not satisfy the security risk constraint, so the available security risk subset is

$$\begin{cases} \mathbf{S}_2^2 = \{\delta_{32}, \delta_{42}, \delta_{82}\}, \\ \mathbf{S}_2^3 = \{\delta_{13}, \delta_{23}, \delta_{53}, \delta_{73}\}. \end{cases}$$

Based on solution step (3), to circularly verify all the satisfied combinations of $m_{c,t}^j$ and $m_{d,t}^j$, and according to the four designed objective optimization function $f^*(m_{c,t}^j, m_{d,t}^j)$, we can determine that the optimal combinations are $(m_{c,t}^2 = 2, m_{d,t}^2 = 1)$ and $(m_{c,t}^3 = 1, m_{d,t}^3 = 3)$.

Lastly, according to the above results and solution step (4), we can obtain the optimal airborne sensor nodes allocation results

$$\{\delta_{32,c}, \delta_{42,d}, \delta_{82,c}\}, \{\delta_{13,d}, \delta_{23,d}, \delta_{53,c}, \delta_{73,d}\}.$$

The estimated error covariance traces of target T_2 and T_3 are shown in Figure 8 in the condition of the above allocation options, the estimated values $E[\text{tr}\mathbf{P}_{e,t}^2] \approx 5.6625$ and $E[\text{tr}\mathbf{P}_{e,t}^3] \approx 7.1698$, the system survivability index $s_t = 0.8909$ ($\lambda = 0.08, c_t = 13$), and the value of optimization objective function $f_{optimal\ max}^*(m_{c,t}^{2,3}, m_{d,t}^{2,3}) = 0.9769$.

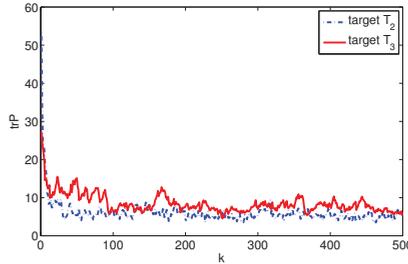


Figure 8. Optimal allocation trace of fusion error covariance of target T_2 and T_3 .

5.3. Analysis of Simulation Results

For the above two different simulation situations, we achieved the optimal allocation of the multi-sensor target tracking fusion system with the optimization model designed in this paper. In time period t , based on the real-time system performance requirements, the optimal number of sensor and optimal fusion method combinations are determined to maximize the system tracking accuracy, and it satisfies the system survivability constraint and other necessary constraints, which can achieve the system self-adaptive performance optimization adjustment function and improve the self-adaptive adjustment ability of the distributed tracking fusion system.

The optimization model in this paper is designed for the problem of system performance instability caused by the changing situations and emergencies in the tracking process. In simulation 1, the system encountered the node failure problem, and in simulation 2, the system encountered the problem of node measurement distance and the security risk of a single airborne plane. From the results of simulation 1 and simulation 2, we can see that the systems have self-adjusting abilities; when the system encounters the above problems and other issues, the system can reject the troublesome

nodes and recombine the remaining system resources to optimally allocate, which leads to the stable performance and great anti-interference ability of the target tracking fusion system.

6. Conclusions

For the problems of a complex motion model in target tracking process and the changeable motion situations which lead to the instability of tracking systems, this paper studies the flexible fusion structure algorithms and designs a kind of flexible fusion optimization model. The multi-step solution strategy and ant colony optimization algorithm are used to solve the designed model, which can obtain the optimal sensor subsets selection dynamically. By the simulation verification, the designed system optimization model was proven to be feasible and effective, and could improve the self-adjustment ability of a target tracking fusion system and guarantee that the multi-plane cooperative tracking task is accomplished successfully.

Acknowledgments: This work was partially supported by Zhejiang Provincial Nature Science Foundation of China (Grant No. LR17F030005), the National Nature Science Fund of China (NSFC) (Grant Nos. 61371064 and 61333011) and opening project of State Key Laboratory of Management and Control for Complex Systems of Institute of Automation of Chinese Academy of Sciences. Xiangfeng Wang is supported by Shanghai YangFan No. 15YF1403400 and NSFC No. 11501210.

Author Contributions: Quanbo Ge: propose the key idea and the main solution, and revise the manuscript; Zhongliang Wei: suggestion on the manuscript organization and English writing, and answer to reviewing comments; Tianfa Cheng: discussion on the problem and solution, mainly writing the manuscript, and programming on simulations; Shaodong Chen: discussion on the manuscript background and simulation; Xiangfeng Wang: discussion and suggestion on optimization modeling

Conflicts of Interest: The authors declare no conflict of interest.

References

- Anitha, R.; Renuka, S.; Abudhahir, A. Multi sensor data fusion algorithms for target tracking using multiple measurements. In Proceedings of the 2013 IEEE International Conference on Computational Intelligence and Computing Research, Enathi, India, 26–28 December 2013; pp. 1–4.
- Fu, J.; Ma, R.; Chai, T. Global Finite-Time Stabilization of a Class of Switched Nonlinear Systems with the Powers of Positive Odd Rational Numbers. *Automatica* **2015**, *54*, 360–373.
- Tharmarasa, R.; Kirubarajan, T.; Sinha, A.; Lang, T. Decentralized sensor selection for large-scale multisensor-multitarget tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 1307–1324.
- Gao, H.; Chen, T. New Results on Stability of Discrete-Time Systems with Time-Varying State Delay. *IEEE Trans. Autom. Control* **2007**, *52*, 328–334.
- Mark, P.K.; Waymond, R.S.; Leslie, M.C. A framework for information-based sensor management for the detection of static target. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **2011**, *41*, 105–120.
- Liu, H.; Guo, D.; Sun, F. Object recognition using tactile measurements: Kernel sparse coding methods. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 656–665.
- Shi, Y.; Han, Z.Z. Adaptive UKF method with applications to target tracking. *Acta Autom. Sin.* **2011**, *37*, 755–759.
- Chen, Z.M.; Bo, Y.M.; Wu, P.L.; Liu, Z.F. Novel particle filter algorithm based on adaptive particle swarm optimization and its application to radar target tracking. *Signal Control* **2013**, *28*, 193–200.
- Ashwin, Y.; Naren, N.; Ananthasayam, M.R.; Gaur, A.; Singh, Y.N. A constant gain kalman filter approach to target tracking in wireless sensor networks. In Proceedings of the 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS), Chennai, India, 6–9 August 2012; pp. 1–7.
- Ma, Z.Q.; Zheng, Y.C.; Liang, J.H.; Li, D.X. An improved kalman filter algorithm for maneuvering target tracking based on ANFIS. *Sens. Transducers* **2013**, *160*, 437–444.
- Sun, J.; Hu, R.; Yang, H. Air target tracking based on improved extended kalman filtering algorithm. *Shipboard Electron. Countermeas.* **2013**, *36*, 34–37.
- Cheng, T.F.; Ge, Q.B.; Shao, T.; Chen, S. Flexible fusion structure for air task networks. In Proceedings of the 2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems, Beijing, China, 28–29 September 2014; pp. 1–6.

13. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422.
14. Wan, E.A.; Vandermerwe, R. The unscented kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, Lake Louise, AB, Canada, 1–4 October 2000; pp. 153–158.
15. Li, W.B.; Ge, Q.B. UKF-STF Tracking with correlated noises for the nonlinear system. In Proceedings of the 2010 8th World Congress on Intelligent Control and Automation, Jinan, China, 7–9 July 2010; pp. 3466–3471.
16. Han, C.Z.; Zhu, H.Y.; Duan, Z.S. *Multi-Source Informance Fusion*; Tsinghua University Press: Beijing, China, 2006.
17. Gao, H.; Chen, T. \mathcal{H}_∞ Estimation for Uncertain Systems With Limited Communication Capacity. *IEEE Trans. Autom. Control* **2007**, *52*, 2070–2084.
18. Yang, H.Y.; You, Z.; Wang, L. Dynamic sensor management for distributed tracking in collaborative network. *J. South China Univ. Technol.* **2012**, *40*, 49–56.
19. Liu, Q.; Liu, Z.; Liu, Y.F. Maneuvering target collaborative tracking algorithm with multi-sensor deployment optimization. *Syst. Eng. Electron.* **2013**, *35*, 304–309.
20. Liu, H.; Liu, Y.H.; Sun, F. Robust exemplar extraction using structured sparse coding. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 1816–1821.
21. Thunemann, P.Z.; Mattikalli, R.; Arroyo, S.; Frank, O. Characterizing the tradeoffs between different sensor allocation and management algorithms. In Proceedings of the 2009 12th International Conference on Information Fusion, Seattle, WA, USA, 6–9 July 2009; pp. 1473–1480.
22. Farina, A.; Ristic, B.; Timmoneri, L. Cramer-rao bound for nonlinear filtering with $P_d < 1$ and its application to target tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 1916–1924.
23. Tharmarasa, R.; Kirubarajan, T.; Hernandezm, L. PCRLB based multisensory array management for multitarget tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 539–555.
24. Lai, J.B.; Wang, H.Q.; Wang, J. Survey of information system survivability. *Comput. Sci.* **2007**, *34*, 237–240.
25. Cheng, T.F.; Ge, Q.B.; Liu, B. Comparison on weights calculation methods based on improved air combat threat assessment model. *Fire Control Command Control China* **2016**, *41*, 32–36.
26. Chen, X.; Kong, Y.Y.; Fang, X.; Wu, Q. A fast two-stage ACO algorithm for robotic path planning. *Neural Comput. Appl.* **2013**, *22*, 313–319.
27. Angelo, J.S.; Bernardino, H.S.; Barbosa, H.J.C. Ant colony approaches for multiobjective structural optimization problems with a cardinality constraint. *Adv. Eng. Softw.* **2015**, *80*, 101–115.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Development of Data Registration and Fusion Methods for Measurement of Ultra-Precision Freeform Surfaces

Ling Bao Kong ¹, Ming Jun Ren ^{2,*} and Min Xu ¹

¹ Shanghai Engineering Research Center of Ultra-Precision Optical Manufacturing, Fudan University, Shanghai 200433, China; LKong@fudan.edu.cn (L.B.K.); minx@fudan.edu.cn (M.X.)

² Institute of Robotics, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

* Correspondence: renmj@sjtu.edu.cn; Tel.: +86-21-3420-4616

Academic Editors: Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang

Received: 23 March 2017; Accepted: 9 May 2017; Published: 12 May 2017

Abstract: The measurement of ultra-precision freeform surfaces commonly requires several datasets from different sensors to realize holistic measurements with high efficiency. The effectiveness of the technology heavily depends on the quality of the data registration and fusion in the measurement process. This paper presents methods and algorithms to address these issues. An intrinsic feature pattern is proposed to represent the geometry of the measured datasets so that the registration of the datasets in 3D space is casted as a feature pattern registration problem in a 2D plane. The accuracy of the overlapping area is further improved by developing a Gaussian process based data fusion method with full consideration of the associated uncertainties in the measured datasets. Experimental studies are undertaken to examine the effectiveness of the proposed method. The study should contribute to the high precision and efficient measurement of ultra-precision freeform surfaces on multi-sensor systems.

Keywords: data fusion; data registration; intrinsic surface features; ultra-precision freeform surfaces; precision metrology

1. Introduction

Ultra-precision freeform surfaces with sub-micrometer-form accuracy and surface finish in the nanometric range are now widely adopted in opto-mechatronic applications, due to their superior mechanical and optical properties in improving the performance of the products in both functionality and size reduction [1]. The trend of miniaturization has further driven the integration of multi-scale representations of features in single surfaces which exhibit specific functionalities at different scales [2,3]. However, the geometric complexity also brings about many challenges for the measurement of these surfaces, especially large-sized and multi-scaled freeform surfaces.

In the past decades, many precision measurement instruments have been developed with respect to specific and varying metrological needs. Many existing precision measurement instruments only possess limited measuring range at one single measurement, and are difficult to perform the high precision measurement of large size with both high resolution and efficiency. This is particularly true for some freeform surfaces with high slopes or large sizes [4,5]. Multi-sensor instruments have been considered as a promising solution for measuring these kinds of surfaces [6]. Several different sensors are integrated into a single instrument to perform cooperative measurements, so as to enhance measurement range and fidelity, while minimizing measurement cost and time. For example, Werth VideoCheck UA 400 [7] integrates an imaging sensor, tactile scanning sensor, and white light sensor

into a single system which is capable of measuring complex 3D geometries with sub-micrometric accuracy. WITec GmbH [8] integrates confocal Raman microscopy, atomic force microscopy, and scanning near-field optical microscopy, and can perform relatively fast measurements of large-area samples. However, these instruments simply integrate several sensors into a common system, and lack the much-required multi-sensor data fusion functionality and characterization that would achieve improved measurement results.

Multi-sensor datasets may come from different spaces with different scales, resolutions, and associated uncertainties. The effectiveness of the multi-sensor metrology heavily depends on the quality of the data registration and fusion, which are further steps for sensor integration, and are responsible for combining the measured datasets from different sensors into a common representation, in order that the measurement can benefit from the technical merits of all the involved sensors. Data registration and fusion has been an emerging technology since the 1990s, and it has previously been used for target tracking, automated identification of targets, and limited automated reasoning applications [9]. The technology is then adopted in reverse engineering and precision metrology [10]. Generally, for surface measurement, the process includes pre-processing, data registration, data fusion and post processing, among which data registration and fusion are the most critical steps [6]. In registration, all the datasets are transformed to a common coordinate frame based on rigid motion. Due to geometry complexity and variety, as well as a lack of common features for freeform surfaces, it is still very difficult to register one freeform surface to another with high precision in the presence of noise. An open literature review shows that data registration can be realized by feature-based or surface description-based approaches, and the registration process generally includes coarse and fine registration [11]. Some research work has been found for data stitching of aspherical surfaces [12,13]. The iterative closest point (ICP) method is often used for correspondence searching [14]. However, this is susceptible to data noise, and outliers are involved in the measured data, and accumulative errors would be produced when a large amount of datasets are involved [15]. There is still little research into data registration and fusion of ultra-precision freeform surfaces with sub-micrometer form accuracy.

Fusion is responsible for processing the redundant data in the overlapping area of the datasets. Considering that the fused datasets may have different resolutions with different associated uncertainties, proper fusion process should be carried out to fuse the datasets which may have different resolutions with different associated uncertainties. Wang et al. [16] reviewed current data fusion methods in surface metrology, and summarized the data fusion methods into four categories, including repeated measurements, stitching, range image fusion, and 3D data fusion. Ramasamy et al. [17] presented several data fusion strategies and weighting methods in the fusion of multi-scaled range images. Although the validity of the method has been confirmed for the measurement of micro-structured surfaces, the uncertainty propagation is not clearly demonstrated. Forbes et al. [18] presented a weighted least square-based data fusion method that relies on linear approximation of the geometry of the datasets. The method may be problematic when the datasets have sharp geometrical changes, for instance, a smooth surface embedded with micro-structures.

This paper presents a study of data registration and fusion for measuring ultra-precision freeform surfaces on multi-sensor instruments. The method performs the data registration by representing the geometry of each dataset based on intrinsic surface features which are invariant to coordinate transformation, and free from the implicit parameterization of the surface. Data fusion is then performed at the overlapping area based on a Gaussian process model, to further reduce the measurement uncertainty. Experimental studies are presented to demonstrate the validity of the proposed method.

2. Data Registration and Fusion Methods

Intrinsic surface features (ISFs) refer to those surface features whose values are invariant under the transformation (rotation/translation) of the embedded coordinate frame, and also free from the implicit parameterization of the surface. Surface registration can be performed without the need to

consider the misalignment of the coordinate frames and the parameterization of the surfaces when the two surfaces are represented by ISFs. Two of the important ISFs are Gaussian and mean curvature. They uniquely determine the surface shape according to the Gaussian Curvature Uniqueness Theorem and the Mean Curvature Uniqueness Theorem [19,20].

Figure 1 shows a flow chart of the proposed data registration and fusion method. Different sets of data were obtained with measurement setup information, and the surface normal/computer aided design (CAD) model was provided. The data format was unified, and hence the ISFs of the data sets were calculated, followed by registration to find the correspondence. The overlapping area among the datasets was then identified, and data fusion was carried out. Figure 2 illustrates the data registration process with the aid of the nominal surface/CAD model. The normal surface or CAD model information was provided, and different sets of data were registered to the normal surface or CAD surface firstly, and hence they were stitched to each other. Figure 2 illustrates data stitching process with the aid of normal surface/CAD model. If there was no CAD information, different sets of data were directly registered to each other by maximizing the similarity of their overlapping areas. Some important algorithms are explained in detail in the coming sections.

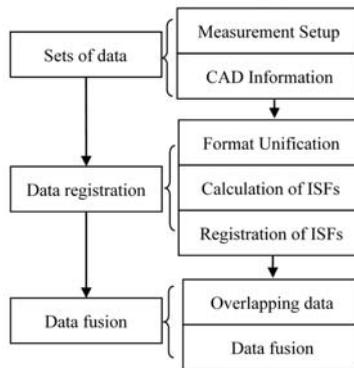


Figure 1. Flow chart of data registration and fusion.

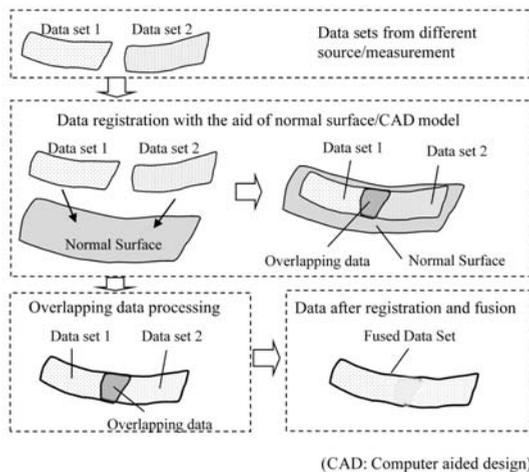


Figure 2. Graphical illustration of data registration and fusion.

2.1. Data Reformat by Re-Sampling

Before registration of the datasets, the format of different datasets was unified by a re-sampling strategy. This was realized by the process as shown in Figure 3. A grid of points were uniformly sampled on a dataset and the values of the ISFs of these points were arranged on a two dimensional (2D) plane to form an intrinsic feature pattern. The pattern was not only invariant to the coordinate transformation but was also free from the implicit parameterization of the surface. The layout of a 2D texture onto a general freeform surface inevitably creates distortion in all but developable surfaces, i.e., surfaces with zero Gaussian curvature, such as a cylinder [19]. Hence, the problem was concerned fitting a 2D pattern into a freeform surface such that the texture distortion was minimized. In the present study, a woven mesh model [21] was employed to address this problem. Woven mesh model is a kind of woven fabric model, which consists of several types of springs in different directions. These springs have their own initial length at which the spring has zero energy. When the mesh model is fitted into a freeform surface, the texture may be distorted and the directions and lengths of the springs are not preserved as compared with the original 2D pattern. This leads to the strain energy. The distortion can then be minimized by minimizing the strain energy in the mesh model, through optimizing the distribution of the points. More details of the woven mesh model can be found in [21]. After that, the format of datasets and normal surface were unified for later registration based on ISF.

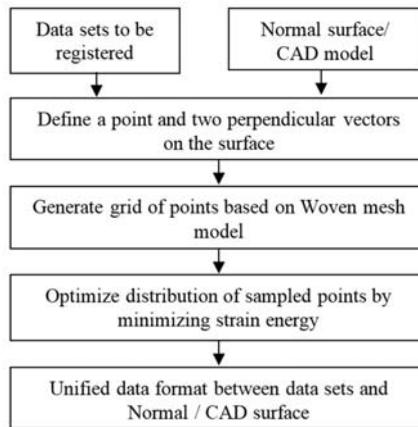


Figure 3. Flow chart for data format unification.

2.2. Calculation of Intrinsic Surface Features

In the present research, Gaussian curvature or mean curvature was chosen as the ISF. Direct calculation of the ISF, such as Gaussian curvature, from the measured datasets is sensitive to the measurement noise. Hence, in the present study, the B-spline surface was used to fit the discrete points so that the measurement noise could be rejected during the surface fitting process and the ISF could be safely calculated [22]. Suppose that $S(u, v)$ is the fitted B-spline surface, where (u, v) are the parameters of the surface; the first fundamental form of $S(u, v)$ can be expressed as

$$I = Edu^2 + 2Fdudv + Gdv^2 \quad (1)$$

where

$$\begin{cases} E = S_u^2 = \left(\frac{\partial S(u,v)}{\partial u} \right)^2 \\ F = S_u S_v = \frac{\partial S(u,v)}{\partial u} \frac{\partial S(u,v)}{\partial v} \\ G = S_v^2 = \left(\frac{\partial S(u,v)}{\partial v} \right)^2 \end{cases} \quad (2)$$

The second fundamental form of $S(u, v)$ is given by

$$II = Ldu^2 + 2Mdudv + Ndv^2 \tag{3}$$

where,

$$\begin{cases} L = \frac{\partial^2 S(u,v)}{\partial u^2} \cdot \frac{S_u \times S_v}{\|S_u \times S_v\|} \\ M = \frac{\partial^2 S(u,v)}{\partial u \partial v} \cdot \frac{S_u \times S_v}{\|S_u \times S_v\|} \\ N = \frac{\partial^2 S(u,v)}{\partial v^2} \cdot \frac{S_u \times S_v}{\|S_u \times S_v\|} \end{cases} \tag{4}$$

Then, Gaussian curvature K and mean curvature H of the surface $S(u, v)$ are determined by the coefficients of the first and second fundamental forms as follows, respectively.

$$K = \det \left(\begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \right) \det \left(\begin{bmatrix} L & M \\ M & N \end{bmatrix} \right) \tag{5}$$

$$H = \frac{1}{2} \text{tr} \left(\begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \right) \det \left(\begin{bmatrix} L & M \\ M & N \end{bmatrix} \right) \tag{6}$$

where operator $\det(\)$ is the determinant of a matrix; operator $\text{tr}(\)$ is the trace of a matrix.

2.3. Registration Process Based on ISF

The surface registration problem was then converted to ISF registration. That is, the corresponding searching in 3D Cartesian coordinate frame by solving six parameters (three translations plus three rotations) was now performed in 2D space by finding three parameters (two translations plus one rotation), as shown in Figure 4. Registration problems involving translation and rotation were recovered by applying a Fourier-Mellin transform and the phase correlation method [23].

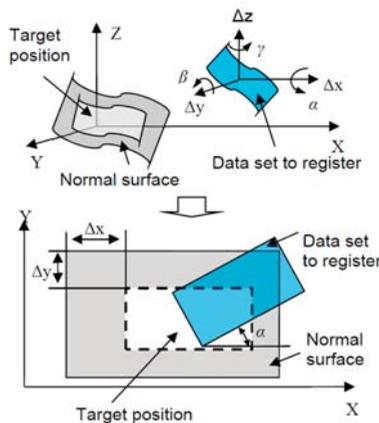


Figure 4. Illustration of data registration based on ISFs.

Suppose $f_2(x, y)$ is translated and rotated from $f_1(x, y)$, then

$$f_2(x, y) = f_1(x \cos(\alpha) - y \sin(\alpha) - \Delta x, x \sin(\alpha) + y \cos(\alpha) - \Delta y) \tag{7}$$

where Δx and Δy are the translation offsets and α is the rotation angle. According to the Fourier translation property and the Fourier rotation property, the Fourier transformation of f_1 and f_2 are related by

$$F_2(\xi, \eta) = \exp(-j2\pi(\xi\Delta x + \eta\Delta y)) \cdot F_1(\xi \cos(\alpha) - \eta \sin(\alpha), \xi \sin(\alpha) + \eta \cos(\alpha)) \quad (8)$$

where F_1 and F_2 are Fourier transform of f_1 and f_2 , respectively. Therefore, Equation (9) preserves

$$M_2(\xi, \eta) = M_1(\xi \cos(\alpha) + \eta \sin(\alpha), -\xi \sin(\alpha) + \eta \cos(\alpha)) \quad (9)$$

where, M_1 and M_2 are magnitudes of F_1 and F_2 , respectively.

From Equation (9), the translation is recovered, and the rotation causes the spectral magnitude to be rotated at the same angle, which can be determined in polar coordinates [24].

$$PM_2(\rho, \theta) = PM_1(\rho, \theta - \alpha) \quad (10)$$

where PM_1 and PM_2 are the spectral magnitudes of f_1 and f_2 in polar coordinates, respectively. The rotating angle can be determined by translation offset in the polar coordinate system by using the phase correlation as follows:

$$Corr(u, v) = \frac{FPM_1(u, v)}{|FPM_1(u, v)|} \cdot \frac{FPM_2(u, v)}{|FPM_2(u, v)|} = \exp(-2\pi(u + v\alpha)) \quad (11)$$

where FPM_1 and FPM_2 are the Fourier transform of PM_1 and PM_2 . The Inverse Fourier Transform of Equation (11) is a Dirac δ -function yielding a sharp maximum at $(0, \alpha)$. Hence, the f_2 is rotated by α , and the rotated f_2 is phase correlated with f_1 again to determine the translational offsets Δx and Δy .

Figure 5 summarizes the algorithms to find the translation and rotation offsets between the two IFPs. It started with inputting two IFPs, i.e., (f_1, f_2) , and the Fourier transform of the two patterns (F_1, F_2) was computed by 2D fast Fourier transform (2D FFT). The spectral magnitudes (M_1, M_2) of the two IFPs were then transformed to the polar coordinate system, and the rotation angle was determined by the phase correlation method. The determined rotation angle α was then used to recover the rotation of the f_2 , and the rotated f_2 , denoted as f_{2r} , was then used to perform phase correlation again with f_1 , to recover the translation.

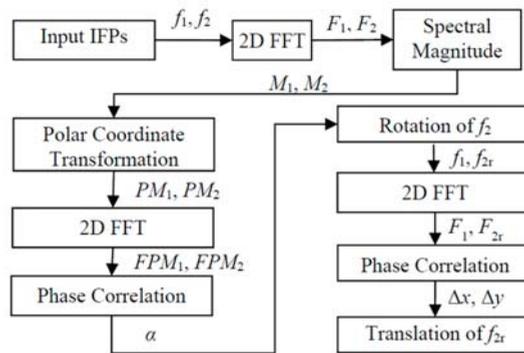


Figure 5. Flowchart of the IFP registration process.

The established correspondence by ISF registration was then used to evaluate the coordinate transformation matrix T by minimizing the sum of the squared distance of each correspondence pairs.

$$F = \sum_{k=0}^n |X1_k - TX2_k|^2 \quad (12)$$

$$T(r_x, r_y, r_z, t_x, t_y, t_z) = \begin{bmatrix} c(r_z)c(r_y) & s(r_z)c(r_y) + c(r_z)s(r_y)s(r_x) & s(r_z)s(r_x) - c(r_z)s(r_y)c(r_x) & t_x \\ -s(r_z)c(r_y) & c(r_z)c(r_x) - s(r_z)s(r_y)s(r_x) & c(r_z)s(r_x) + s(r_z)s(r_y)c(r_x) & t_y \\ s(r_y) & -c(r_y)s(r_x) & c(r_y)c(r_x) & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

where $(X1_k, X2_k)$ are the correspondence pairs, n is the number of the established correspondence pairs, t_x, t_y, t_z are the translation components, and r_x, r_y, r_z are the rotation angles; $c()$ and $s()$ are abbreviations of the cosine and sine functions. The problem can be efficiently solved by the Levenberg-Marquardt algorithm [25].

2.4. Data Fusion Based on Gaussian Process Model

After data registration, the overlapped area of the two registered datasets were processed to form one fused dataset. Challenges exist for overlapped data fusion when the data sets are from different sources (e.g., different sensors), due to the variety of data density and uncertainty involved. In the current study, the Gaussian process (GP) model [26] was used to perform the data fusion by taking the uncertainty of the datasets into account. GP is a Bayesian regression model which can be completely specified by a mean function $\mu(X)$ and covariance function $K(X, X)$, as given by Equation (14).

$$f(X) \sim N(\mu(X), K(X, X)) \quad (14)$$

In the actual measurement, the zero-offset mean function was used, since no prior knowledge on the surface geometry is available. A prediction of f^* at arbitrary location x^* can then be given from the joint distribution of f^* with Z as given by Equation (15) as follows:

$$\begin{bmatrix} Z \\ f^* \end{bmatrix} \sim N \left(0, \begin{bmatrix} K(X, X) + \sigma_\varepsilon^2 I & K(X, x^*) \\ K(X, x^*) & K(x^*, x^*) \end{bmatrix} \right) \quad (15)$$

where I is the identity matrix and σ_ε is a hyperparameter representing the noise variance associated in Z . A prediction m and its uncertainty cov at an arbitrary location x on the model can then be obtained from the marginal distribution of $f(x)$ as follows [27]:

$$m = K(x, X) \left(K(X, X) + \sigma_\varepsilon^2 I \right)^{-1} Z \quad (16)$$

$$cov = K(x, x) - K(x, X) \left(K(X, X) + \sigma_\varepsilon^2 I \right)^{-1} K(X, x) \quad (17)$$

In the present study, the squared exponential function was used as the covariance function that provided correlation among any set of outputs. More details regarding the GP modelling, are contained in the work by Rasmussen et al. [26].

It was assumed that all the datasets at the overlapping areas possessed the same shape and hence the same GP covariance except the noise parameters, since different datasets would have different levels of uncertainties. Then, for the two datasets Z_1 and Z_2 , the fused GP model was simply established by treating the fusion process as a standard GP regression problem as given by Equation (18):

$$\begin{bmatrix} \bar{Z} \\ f_f^* \end{bmatrix} \sim N \left(0, \begin{bmatrix} K(\bar{X}, \bar{X}) + \text{diag}(\sigma_{\varepsilon_1}^2 I, \sigma_{\varepsilon_2}^2 I) & K(\bar{X}, x_f^*) \\ K(\bar{X}, x_f^*) & K(x_f^*, x_f^*) \end{bmatrix} \right) \quad (18)$$

where f_f^* and x_f^* are the prediction and its location, $\bar{Z} = [Z_1, Z_2]$ are the two measured datasets, $\bar{X} = [X_1, X_2]$ are the location of the measured datasets, and $\sigma_{\varepsilon_1}^2$ and $\sigma_{\varepsilon_2}^2$ are the hyperparameters representing the noise variance associated in Z_1 and Z_2 . Recalling that Equation (18) is an extension of Equation (15) that the measured data contains two different levels of noise, hence the mean m_f^* and the variance $\text{cov}(f_f^*)$ of the f_f^* on the fused GP model can also be obtained in similar way by the marginal distribution of f_f^* as given by Equation (19).

$$f_f^* | x_f^*, \bar{X}, \bar{Z} \sim N(m_f^*, \text{cov}(f_f^*)) \quad (19)$$

3. Experimental Verification

3.1. Simulation Study

Simulation studies were undertaken to verify the proposed method for data registration and fusion. A normal surface is defined as

$$\begin{cases} f(x, y) = \sin(x) + \cos(y) \\ x \in [-5, 5], y \in [-3, 3] \end{cases} \quad (20)$$

As shown in Figure 6, two sub-surfaces were sampled from the designed surface with different spacings (0.4 mm and 0.1 mm) at different locations, and were denoted as Surface 1 and Surface 2. Surface 1 was extracted at $x \in [-4.5, 1]$, $y \in [-2, 2]$ and was moved to a position based on the transformation of $T(\frac{\pi}{20}, -\frac{\pi}{15}, \frac{\pi}{30}, -1.5, 0.5, 4)$, based on Equation (13). Surface 2 was extracted at $x \in [-1, 4.5]$, $y \in [-2, 2]$ and was moved to a position based on the transformation of $T(\frac{\pi}{25}, \frac{\pi}{10}, \frac{\pi}{30}, -1.5, -0.5, 5)$. Surface 1 and Surface 2 had Gaussian noise added with standard deviations of 0.2 μm and 0.5 μm respectively, to represent measurement errors. It was noted that Surface 1 and Surface 2 had different resolutions with different associated uncertainties, and were embedded in different coordinate frames, which is very common in multi-sensor surface metrology in order to balance measurement efficiency and accuracy. The proposed method was then used to perform registration and fusion of the two surfaces.

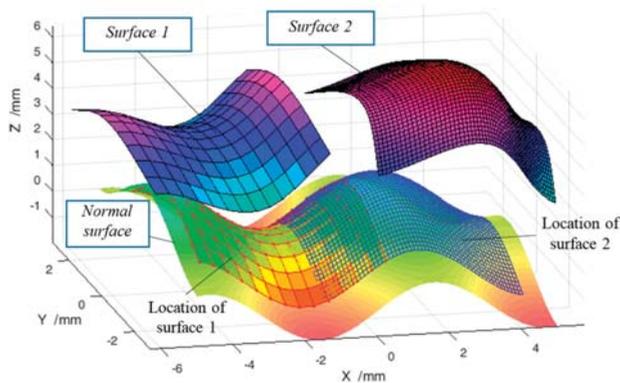


Figure 6. Normal surface and two extracted surfaces with different positions and postures.

Gaussian curvatures were employed as ISFs, and were calculated as shown in Figure 7. No matter where and what positions and postures the sub-surfaces are, their Gaussian curvatures remained the same. This validated that the Gaussian curvature was the ISF of the surface and is free from the coordinate frame. For the next step, surface registration was undertaken based on such ISFs. Figure 8 shows the results of registration of Gaussian curvatures of two sub-surfaces to the normal surface; while Figure 9 is the corresponding surface registration results of the two surfaces to the normal surface data. The registration process was repeated 50 times to evaluate the reliability of the proposed method. Table 1 summarizes the error of the evaluated six spatial parameters for Surface 1, by the proposed method, as well as by the classical ICP method [14]. It was interesting to note from the results that the performance of the proposed method well matched with that of the ICP method, and possessed slightly lower variance. This was due to the fact that the surface reconstruction process not only rejected the noise of the registered datasets, but also created a larger number of correspondences by ISF registration.

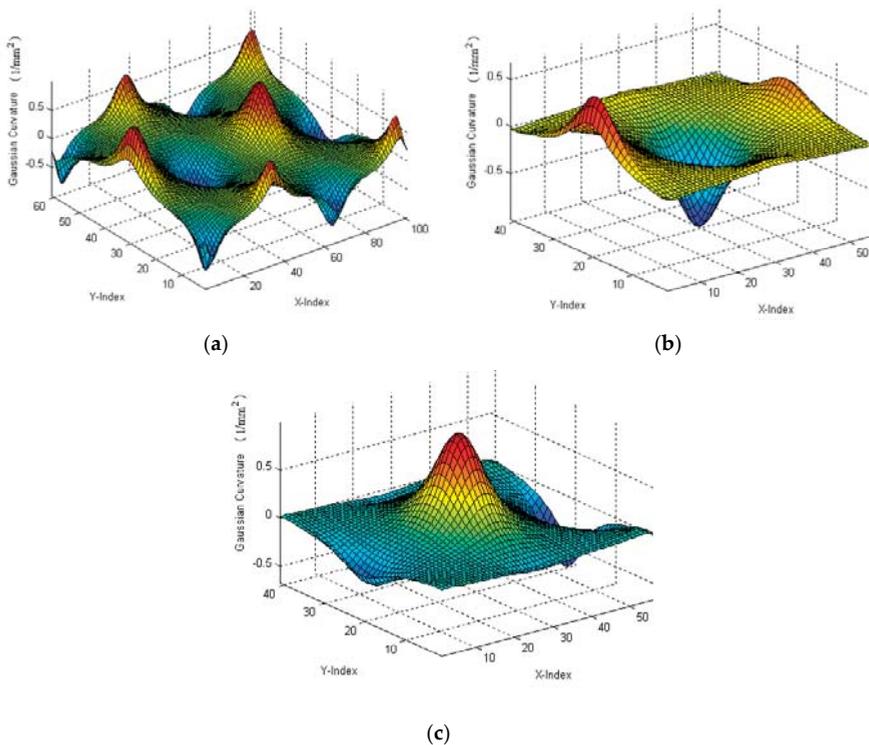


Figure 7. Gaussian curvatures of (a) normal surface; (b) extracted Surface 1; and (c) extracted Surface 2.

Table 1. A summary of the error of the evaluated six spatial parameters.

	r_x (μrad)	r_y (μrad)	r_z (μrad)	t_x (nm)	t_y (nm)	t_z (nm)
ISFM	1.7/0.5 *	4.2/1.9	3.9/1.2	9.9/3.2	7.8/1.9	3.5/0.8
ICPM	2.1/0.6	5.5/2.2	3.7/1.6	12.8/6.3	9.8/2.5	3.4/0.8

* a/b : a refers to mean error, b refers to the variance. ISFM: Intrinsic surface features based method. ICPM: Iterative closest point method.

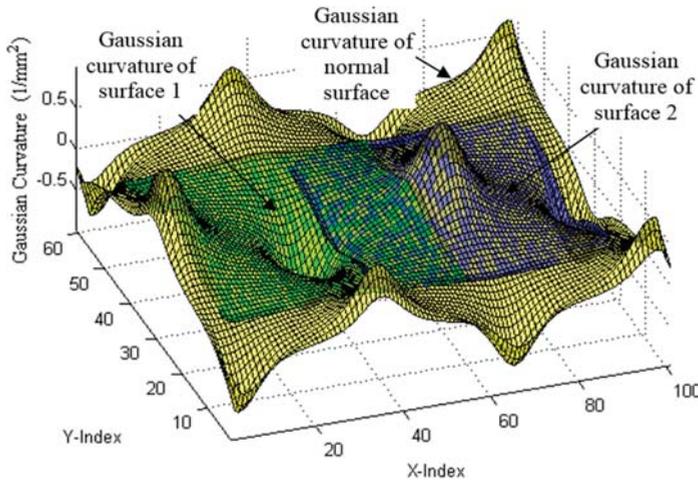


Figure 8. Registration of Gaussian curvatures of two sub-surfaces to normal surface.

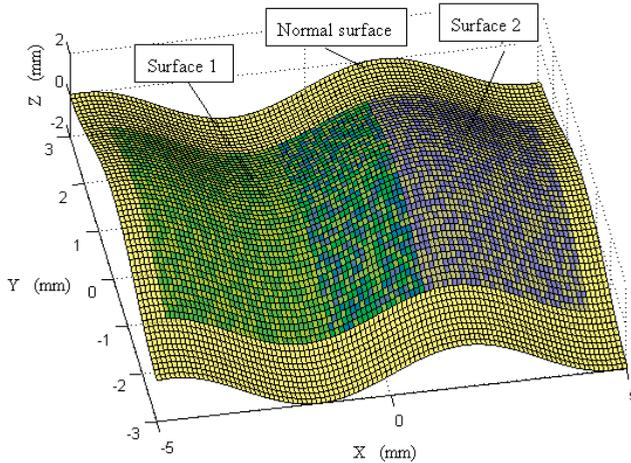


Figure 9. Surface registration results of the two sub-surfaces to normal surface.

After transforming the two surfaces into a common coordinate system, data fusion was then performed at the overlapping area based on the method presented in Section 2.4. It was emphasized that, since the proposed method performed the registration based on the ISFs, the overlapping of the two surfaces could also be easily identified in the ISF registration process. Figure 10 shows the fused GP model and its estimated uncertainty at the overlapping area. It was seen from the estimated uncertainty that the contained noise in the data was successfully estimated in the GP modelling and fusion process. Figure 11 shows the deviation of the established GP model from the designed surface, and Table 2 summarizes the evaluated peak-to-valley (PV), height error, and the root-mean-square (RMS) height error. Since no form error was added to the sampled surface, i.e., Surface 1 and Surface 2, the theoretical form error at the overlapping area should be zero. Hence, the evaluated form error should be the error resulting from the measurement noise. It was seen from the results that, based on the proposed method, the error was reduced to several nanometers via the GP modeling and

fusion process, when the magnitude of the measurement noise was at sub-micrometer level. It was clearly seen from the Table 2 that the proposed method had much better accuracy than the existing method. This means that the proposed method successfully registered the two freeform surfaces under the existence of the measurement noise, and the accuracy of the overlapping area was further improved, based on the proposed GP based data fusion method. The proposed method was also compared with an existing method which was currently widely used in practice. The existing method uses ICP to register the two surfaces, and the data at the overlapping area is fused based on the weighted mean (WM) method [16]. It is interesting to note from the comparison, that the proposed method had a much better performance than the ICP-WM method, in registering and fusing the freeform surfaces under the existence of the measurement noise.

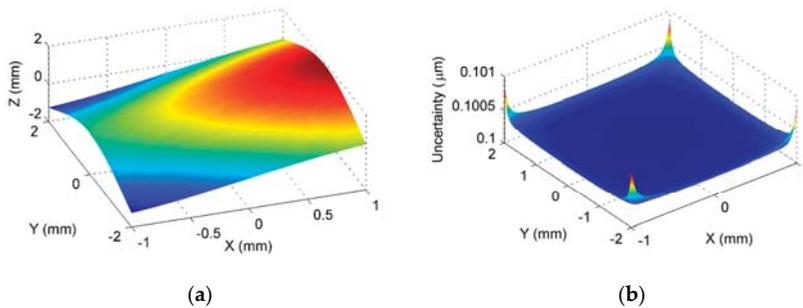


Figure 10. Fusion model of the overlapping area based on the Gaussian process (GP) model. (a) GP model at overlapping area; (b) Estimated uncertainty of the established GP model.

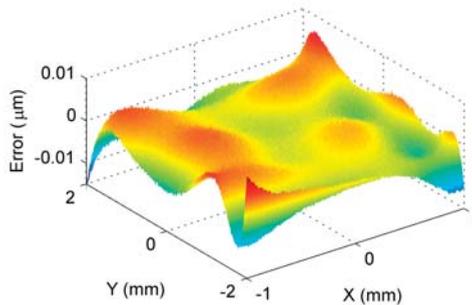


Figure 11. Deviation of the stitched surfaces from the designed surface.

Table 2. A comparison of the ISF+GP method with existing method.

	Fusion by ISF + GP	Fusion by ICP + WM
RMS (nm)	2	7
PV (nm)	24	78

RMS: Root mean square. PV: Peak-to-valley. WM: Weighted mean. ISF: Intrinsic surface features. ICP: Iterative closest point.

3.2. Application in Actual Measurement

An experimental study was conducted to evaluate the performance of the proposed method. A sinusoidal micro-structured surface defined by Equation (21) was machined by a four-axis ultra-precision machining system (Moore Nanotech 350). To fully capture the geometric information,

including the surface form and texture, the workpiece was measured by a 3D optical profiler (Zygo NexView) with two different objectives i.e., $5\times$ and $20\times$ magnifications (zoom was $2.0\times$). The field of view of the two objectives were $0.76 \times 0.76 \text{ mm}^2$ and $0.21 \times 0.21 \text{ mm}^2$, and the maximum slopes of the two objectives were 7.27° and 21.80° , respectively. The measured data are shown in Figure 12.

$$z = 0.015 * (\sin(15x) + \cos(15y)) \quad (21)$$

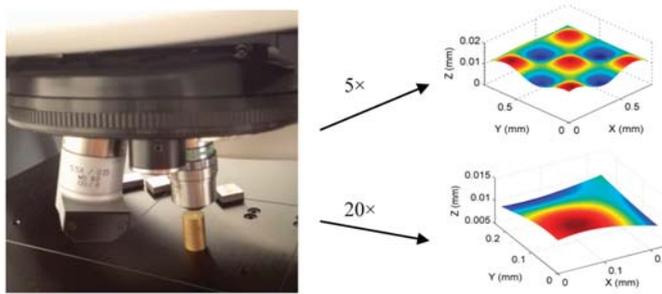


Figure 12. Measurement of a machined micro-structured surface.

The two datasets had different resolutions and were embedded in two different coordinate frames, since they were measured by two different objectives in two steps. As a result, the proposed method was used to perform data stitching and fusion, so as to obtain a unique representation of the machined surface. It started by registering the dataset measured by the $5.5\times$ and $20\times$ objectives to the nominal surface, using the proposed ISF based data stitching method. Filtering was carried out to remove outliers and noise before the calculation of the intrinsic surface feature. Figure 13a shows the registration of the ISFs of the two datasets, and Figure 13b shows the registered two datasets. After data registration, the overlapped area between the two datasets was processed based on the proposed fusion method. To examine the quality of the result, comparison was made between the evaluated form error of the overlapped area on the datasets obtained by $5.5\times$ and $20\times$ objectives, and the fused dataset. Figure 13c shows the form error evaluated by the fused dataset. The RMS errors evaluated by the three datasets were $0.011 \mu\text{m}$, $0.097 \mu\text{m}$, and $0.087 \mu\text{m}$ respectively. It is interesting to note from the results that the RMS errors obtained by fused dataset matched well with those obtained by the $5.5\times$ and $20\times$ objectives, which implied that the registration and fusion had been carried out accurately.

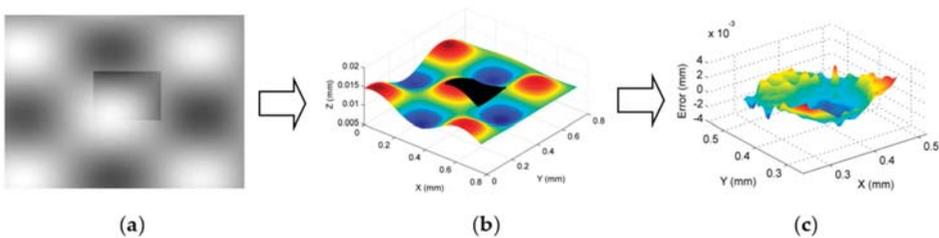


Figure 13. Datasets registration and fusion based on the proposed method. (a) ISF registration; (b) Datasets after registration; (c) Evaluated form error.

To further evaluate the capability of the proposed method, an actual measurement was conducted on another general freeform surface. Figure 14 shows the machined freeform surface which was

designed by the peak function and contained several peaks and valleys, and was a representative type of freeform surface.

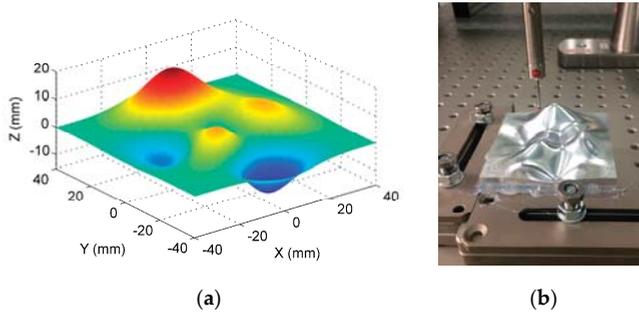


Figure 14. Measurement of a general freeform surface. (a) Designed freeform surface; (b) Machined surface on a coordinate measuring machine (CMM).

The combined use of large scale fast inspection sensors, e.g., laser scanner and photogrammetry, with micro scale pointwise measuring sensors, e.g., coordinate measuring machines, are currently the most common scenarios for multi-sensor surface metrology. Therefore, in the present study, a high precision coordinate measuring machine (CMM) and a laser scanner were combined to measure the freeform surface for efficient and reliable measurement. The CMM possessed a length measurement uncertainty with $U = (0.6 + L/500, L \text{ in mm}) \mu\text{m}$, and a probing error with $u = 0.9 \mu\text{m}$ (1σ , normal). The uncertainty of the laser scanner was identified to be $u = 3.4 \mu\text{m}$ (1σ , normal) by a reference ball. The measurement was carried out in two steps. Firstly, CMM was used to measure the surface with spacing 1 mm in both X and Y directions. A total of 6456 points were uniformly sampled with 1 mm spacing over the entire surface. Figure 15 shows the form error evaluation results. The PV and RMS values of the measured surface were found to be $34.9 \mu\text{m}$ and $5.5 \mu\text{m}$. The values served as a benchmark to verify the effectiveness of the proposed method.

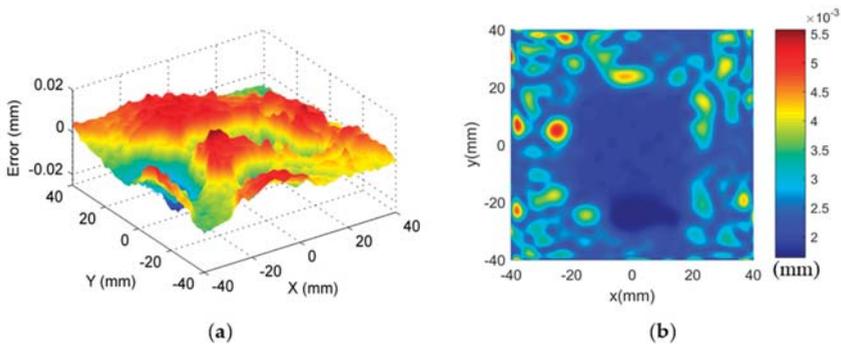


Figure 15. Form error evaluation results. (a) Evaluated form error; (b) Uncertainty of the fused GP model.

Secondly, multi-sensor measurement strategy was carried out. CMM was used to measure the surface with 4 mm spacing in both X and Y directions. Laser scanning was used to perform dense measurements of the surface. The proposed method was then used to process the measured datasets. ISF-based data registration method was used to precisely register the measured datasets by the two

sensors into a common coordinate system. The datasets at the identified overlapping area were then fused based on the GP model. To verify the effectiveness of the proposed method, only the overlapping part was used for the form error evaluation of the machined surface. A summary of the experimental results is shown in Table 3. Both the evaluation parameters and the time consumption for the measurement are given to analyze both the efficiency and accuracy of the proposed method.

Table 3. A summary of the experimental results.

	PV	RMS	Time (h)
Benchmarking	36.8	5.7	>3
Laser scanner	49.6	8.1	<0.16
CMM	33.2	4.8	~0.5
Hybrid	35.4	5.5	~0.5

As a fast measurement sensor, the laser scanner has the highest measurement efficiency, while its accuracy is relatively low compared with CMM. For CMM measurements with loose sampling, the measurement time became acceptable, while the inadequate sampling of the surface would underestimate the form error of the machined surface, especially the PV, which was determined by extreme points. Hybrid measurement on the other hand, had the best overall performance. It was shown from results that, by registering and fusing the datasets from the laser scanner and the CMM, accuracy was dramatically improved, while the measurement efficiency was maintained as well. The experiment verified the capability of the multi-sensor measurement in freeform surface measurement, and further confirmed the effectiveness of the proposed method in addressing the key issues in the measurement process.

4. Conclusions

Freeform data stitching and fusion technology provides a practical solution for multi-sensor measurement of freeform surfaces, and also for enhancing the measuring ability of some high precision measurement instruments. This paper presented methods and algorithms for data stitching and fusion for measuring ultra-precision freeform surfaces based on the registration of ISFs. Some important algorithms involved in the registration algorithms were explained, including data format unification, calculation of ISFs such as Gaussian curvatures and mean curvatures, registration of the ISF map, Gaussian process-based data fusion for the overlapping area, etc. Experimental studies were conducted and the results were discussed. The proposed method and algorithms of data stitching and fusion intend to eliminate or alleviate the effect of noise and outliers, and provide a robust registration and fusion method, which is helpful for multi-scale and multi-sensor measurement of ultra-precision freeform surfaces.

Acknowledgments: The work was substantially supported by the National Natural Science Foundation of China (Grant No. 51505404), Science Challenging Program of CAEP (Grand No. JCKY2016212A506-0106) and National Key Instrument Project (Grand No. 2016YFF0102003-02).

Author Contributions: Ling Bao Kong led the conception, design and writing of the manuscript; Ming Jun Ren developed the methods and algorithms, Min Xu contributed to the experimental work and writing.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brinksmeier, E.; Schonemann, L. Generation of discontinuous microstructures by diamond micro chiseling. *CIRP Ann. Manuf. Technol.* **2014**, *63*, 49–52. [CrossRef]
2. Jiang, X.; Whitehouse, D.J. Technological shifts in surface metrology. *CIRP Ann. Manuf. Technol.* **2012**, *61*, 815–836. [CrossRef]

3. Whitehouse, D.J. Surface geometry, miniaturization and metrology. *Phil. Trans. R. Soc. A* **2012**, *370*, 4042–4065. [CrossRef] [PubMed]
4. Leach, R.; Boyd, R.; Burke, T.; Danzebrink, H.U.; Dirscherl, K.; Dziomba, T.; Gee, M.; Koenders, L.; Morazzani, V.; Pidduck, A.; et al. The European nanometrology landscape. *Nanotechnology* **2011**, *22*, 062001. [CrossRef] [PubMed]
5. Weckenmann, A.; Peggs, G.; Hoffmann, J. Probing systems for dimensional micro- and nano-metrology. *Meas. Sci. Technol.* **2006**, *17*, 504–509. [CrossRef]
6. Weckenmann, A.; Jiang, X.; Sommer, K.; Newschaefer-Rube, U.; Seewig, J.; Shaw, L.; Estler, T. Multisensor data fusion in dimensional metrology. *CIRP Ann. Manuf. Technol.* **2009**, *58*, 701–721. [CrossRef]
7. Werth Messtechnik GmbH. Available online: <http://www.werthmesstechnik.de> (accessed on 22 March 2017).
8. Witec GmbH. Available online: <http://www.witec.de/products/> (accessed on 22 March 2017).
9. Hall, D.L.; Linas, J. An Introduction to Multisensor Data Fusion. *Proc. IEEE* **1997**, *5*, 6–23. [CrossRef]
10. Yau, H.T.; Kuo, C.C.; Yeh, C.H. Extension of surface reconstruction algorithm to the global stitching and repairing of STL models. *Comput. Aided Des.* **2003**, *35*, 477–486. [CrossRef]
11. Marinello, F.; Bariani, P.; De Chiffre, L.; Hansen, H.N. Development and analysis of a software tool for stitching three-dimensional surface topography data sets. *Meas. Sci. Technol.* **2007**, *18*, 1404–1412. [CrossRef]
12. Zhao, C.; Burge, J.H. Stitching of off-axis sub-aperture null measurements of an aspheric surface. *Proc. SPIE* **2008**, 706316. [CrossRef]
13. Wang, X.; Wang, L.; Yin, L.; Zhang, B.; Fan, D.; Zhang, X. Measurement of large aspheric surfaces by annular subaperture stitching interferometry. *Chin. Opt. Lett.* **2007**, *5*, 645–647.
14. Besl, P.J.; McKay, N.D. A method for registration of 3D shapes. *IEEE Trans. Pattern Anal.* **1992**, *14*, 239–256. [CrossRef]
15. Ren, M.J.; Cheung, C.F.; Kong, L.B. Invariant-Feature-Pattern-Based Form Characterization for the Measurement of Ultraprecision Freeform Surfaces. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 963–973. [CrossRef]
16. Wang, J.; Leach, R.; Jiang, X. Review of the mathematical foundations of data fusion techniques in surface metrology. *Surf. Topogr. Metrol. Prop.* **2015**, *3*, 023001. [CrossRef]
17. Ramasamy, S.K. Multi-scale Data Fusion for Surface Metrology. Ph.D. Thesis, The University of North Carolina, Charlotte, NC, USA, 2011.
18. Forbes, A.B. Weighting observations from multi-sensor coordinate measuring systems. *Meas. Sci. Technol.* **2012**, *23*, 025004. [CrossRef]
19. Horn, B.K.P. Extended Gaussian images. *Proc. IEEE* **1984**, *72*, 1656–1678. [CrossRef]
20. Hsiung, C.C. *A First Course in Differential Geometry*; Wiley-Interscience: New York, NY, USA, 1981.
21. Wang, C.L.; Tang, K.; Yeung, M.L. Freeform surface flattening based on fitting a woven mesh model. *Comput. Aided Des.* **2005**, *37*, 799–814. [CrossRef]
22. Ren, M.J.; Cheung, C.F.; Kong, L.B. A robust surface fitting and reconstruction algorithm for form characterization of ultra-precision freeform surfaces. *Measurement* **2011**, *44*, 2068–2077. [CrossRef]
23. Chen, Q.; Defrise, M.; Deconinck, F. Symmetric Phase-only matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 1156–1168. [CrossRef]
24. Takita, K.; Aoki, T.; Sasaki, Y.; Higuchi, T.; Kobayashi, K. High-accuracy subpixel image registration based on phase-only correlation. *IEICE Trans. Fundam.* **2003**, *E86-A*, 1925–1934.
25. Marquardt, D.W. An algorithm for least squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441. [CrossRef]
26. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
27. Roberts, S.; Osborne, M.; Ebdon, M.; Reece, S.; Gibson, N.; Aigrain, S. Gaussian processes for time-series modelling. *Philos. Trans. A Math. Phys. Eng. Sci.* **2013**, *371*, 20110550. [CrossRef] [PubMed]



Article

Optimal Fusion Estimation with Multi-Step Random Delays and Losses in Transmission

Raquel Caballero-Águila ^{1,*}, Aurora Hermoso-Carazo ² and Josefa Linares-Pérez ²

¹ Dpto. de Estadística, Universidad de Jaén, Paraje Las Lagunillas, 23071 Jaén, Spain

² Dpto. de Estadística, Universidad de Granada, Avda. Fuentenueva, 18071 Granada, Spain; ahermoso@ugr.es (A.H.-C.); jlinares@ugr.es (J.L.-P.)

* Correspondence: raguila@ujaen.es; Tel.: +34-953-212-926

Academic Editors: Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang

Received: 7 April 2017; Accepted: 15 May 2017; Published: 18 May 2017

Abstract: This paper is concerned with the optimal fusion estimation problem in networked stochastic systems with bounded random delays and packet dropouts, which unavoidably occur during the data transmission in the network. The measured outputs from each sensor are perturbed by random parameter matrices and white additive noises, which are cross-correlated between the different sensors. Least-squares fusion linear estimators including filter, predictor and fixed-point smoother, as well as the corresponding estimation error covariance matrices are designed via the innovation analysis approach. The proposed recursive algorithms depend on the delay probabilities at each sampling time, but do not need to know if a particular measurement is delayed or not. Moreover, the knowledge of the signal evolution model is not required, as the algorithms need only the first and second order moments of the processes involved. Some of the practical situations covered by the proposed system model with random parameter matrices are analyzed and the influence of the delays in the estimation accuracy are examined in a numerical example.

Keywords: recursive fusion estimation; sensor networks; random parameter matrices; random delays; packet dropouts

1. Introduction

Over the last few decades, research on the estimation problem for networked stochastic systems has gained considerable attention, due to the undeniable advantages of networked systems, whose applicability is encouraged, among other causes, by the development and advances in communication technology and the growing use of wireless networks. As it is well known, the Kalman filter provides a recursive algorithm for the optimal least-squares estimator in stochastic linear systems, assuming that the system model is exactly known and all of the measurements are instantly updated. The development of sensor networks motivates the necessity of designing new estimation algorithms that integrate the information of all the sensors to achieve a satisfactory performance; thus, using different fusion techniques, the measurements from multiple sensors are combined to obtain more accurate estimators than those obtained when a single sensor is used. In this framework, important extensions of the Kalman filter have been proposed for conventional sensor networks in which the measured outputs of the sensors always contain the actual signal contaminated by additive noises, and the transmissions are carried through perfect connections (see, e.g., [1–3] and references therein).

However, in a network environment, usually the standard observation models are not suitable due to the existence of network-induced uncertainties that can occur in both of the sensor measured outputs, and during the data transmission through the network. Accordingly, the consideration of appropriate observation models is vitally important to address the estimation problem in networked systems. Random failures in the transmission of measured data, together with the inaccuracy of

the measurement devices, cause often the degradation of the estimator performance in networked systems. In light of these concerns, the estimation problem with one or even several network-induced uncertainties is recently attracting considerable attention, and the design of new fusion estimation algorithms has become an active research topic (see, e.g., [4–13] and references therein). In addition, some recent advances on the estimation, filtering and fusion for networked systems with network-induced phenomena can be reviewed in [14,15], where a detailed overview of this field is presented.

One of the most common network-induced uncertainties in the measured outputs of the different sensors is the presence of multiplicative noise, due to different reasons, such as interferences or intermittent sensor failures. Specifically, in situations involving random observation losses (see [5]), sensor gain degradation (see [6]), missing or fading measurements (see [13,16], respectively), the sensor observation equations include multiplicative noises. A unified framework to model these random phenomena is provided by the use of random measurement matrices in the sensor observation model. For this reason, the estimation problem in networked systems with random measurement matrices has become a fertile research subject, since this class of systems allow for covering different networked-induced random uncertainties as those mentioned above (see, e.g., [17–23], and references therein).

In relation to the network-induced uncertainties during the data transmission, it must be indicated that sudden changes in the environment and the unreliability of the communication network, together with the limited bandwidths of communication channels, cause unavoidable random failures during the transmission process. Generally, random communication delays and/or transmission packet dropouts are two essential issues that must be taken into account to model the measurements, which, being available after transmission, will be used for the estimation. Several estimation algorithms have been proposed in multisensor systems considering either transmission delays or packet losses (see, e.g., [24,25]) and also taking into account random delays and packet dropouts simultaneously (see, e.g., [4,23,26]). By using the state augmentation method, systems with random delays and packet dropouts can be transformed into systems with random parameter matrices (see, e.g., [8–10,20]). Hence, systems with random parameter measurement matrices also provide an appropriate unified context for modelling these random phenomena in the transmission.

Nevertheless, it must be indicated that the state augmentation method leads to a rise of the computational burden, due to the increase of the state dimension. Actually, in models with more than one or two-step random delays, the computational cost can be excessive and alternative ways to model and address the estimation problem in this class of systems need to be investigated. Recently, a great variety of models have been used to describe the phenomena of multi-step random delays and packet losses during the data transmission in networked systems, and fusion estimation algorithms have been proposed based on different approaches—for example, the recursive matrix equation method in [6], the measurement reorganisation approach in [27], the innovation analysis approach in [28] and the state augmentation approach in [29–31]. It should be noted that, in the presence of multi-step random delays and packet losses during the data transmission, many difficulties can arise in the design of the optimal estimators when the state augmentation approach is not used.

In view of the above considerations, this paper is concerned with the optimal fusion estimation problem, in the least-squares linear sense, for sensor networks featuring stochastic uncertainties in the sensor measurements, together with multi-step random delays and packet dropouts during the data transmission. The derivation of the estimation algorithms will be carried out without using the evolution model generating the signal process. The uncertainties in the measured outputs of the different sensors are described by random measurement matrices. The multi-step random delays in the transmissions are modeled by using a collection of Bernoulli sequences with known distributions and different characteristics at each sensor; the exact value of these Bernoulli variables is not required, and only the information about the probability distribution is needed. To the best of the authors' knowledge, the optimal estimation problem (including prediction, filtering and fixed-point smoothing) has not

been investigated for systems involving random measurement matrices and transmission multi-step random delays simultaneously, and, therefore, it constitutes an interesting research challenge. The main contributions of this research can be highlighted as follows: (a) even though our approach, based on covariance information, does not require the signal evolution model, the proposed algorithms are also applicable in situations based on the state-space model (see Remark 1); (b) random measurement matrices are considered in the measured outputs, thus providing a unified framework to address different network-induced phenomena (see Remark 2); (c) besides the stochastic uncertainties in the sensor measurements, simultaneous multi-step random delays and losses with different rates are considered in the data transmission; (d) unlike most papers about multi-step random delays, in which only the filtering problem is considered, we propose recursive algorithms for the prediction, filtering and fixed-point smoothing estimators under the innovation approach, which are computationally very simple and suitable for online applications; and (e) optimal estimators are obtained without using the state augmentation approach, thus reducing the computational cost in comparison with the augmentation method.

The rest of the paper is organized as follows. In Section 2, we present the sensor network and the assumptions under which the optimal linear estimation problem will be addressed. In Section 3, the observation model is rewritten in a compact form and the innovation approach to the least-squares linear estimation problem is formulated. In Section 4, recursive algorithms for the prediction, filtering and fixed-point smoothing estimators are derived. A simulation example is given in Section 5 to show the performance of the proposed estimators. Finally, some conclusions are drawn in Section 6.

Notations. The notations used throughout the paper are standard. \mathbb{R}^n and $\mathbb{R}^{m \times n}$ denote the n -dimensional Euclidean space and the set of all $m \times n$ real matrices, respectively. For a matrix A , A^T and A^{-1} denote its transpose and inverse, respectively. The shorthand $Diag(A_1, \dots, A_m)$ stands for a block-diagonal matrix whose diagonal matrices are A_1, \dots, A_m . $\mathbf{1}_n = (1, \dots, 1)^T$ denotes the all-ones $n \times 1$ -vector and I_n represent the $n \times n$ -identity matrix. If the dimensions of matrices are not explicitly stated, they are assumed to be compatible with algebraic operations. The Kronecker and Hadamard product of matrices will be denoted by \otimes and \circ , respectively. $\delta_{k,s}$ denotes the Kronecker delta function. For any $a, b \in \mathbb{R}$, $a \wedge b$ is used to mean the minimum of a and b .

2. Observation Model and Preliminaries

In this paper, the optimal fusion estimation problem of multidimensional discrete-time random signals from measurements obtained by a sensor network is addressed. At each sensor, the measured outputs are perturbed by random parameter matrices and white additive noises that are cross-correlated at the same sampling time between the different sensors. The estimation is performed in a processing centre connected to all sensors, where the complete set of sensor data is combined, but due to eventual communication failures, congestion or other causes, random delays and packet dropouts are unavoidable during the transmission process. To reduce the effect of such delays and packet dropouts without overloading the network traffic, each sensor measurement transmits a fixed number of consecutive sampling times and, when several packets arrive at the same time, the receiver discards the oldest ones, so that only one measured output is processed for the estimation at each sampling time.

2.1. Signal Process

The optimal estimators will be obtained using the least-squares (LS) criterion and without requiring the evolution model generating the signal process. Actually, the proposed estimation algorithms, based on covariance information, only need the mean vectors and covariance functions of the processes involved, and the only requirement will be that the signal covariance function must be factorizable according to the following assumption:

(A1) The n_x -dimensional signal process $\{x_k; k \geq 1\}$ has zero mean and its autocovariance function is expressed in a separable form, $E[x_k x_s^T] = A_k B_s^T, s \leq k$, where $A_k, B_s \in \mathbb{R}^{n_x \times n}$ are known matrices.

Remark 1 (on assumption (A1)). *The estimation problems based on the state-space model require new estimation algorithms when the signal evolution model is modified; therefore, the algorithms designed for stationary signals driven by $x_{k+1} = \Phi x_k + \zeta_k$ cannot be applied for non-stationary signals generated by $x_{k+1} = \Phi_k x_k + \zeta_k$, and these, in turn, cannot be used in uncertain systems where $x_{k+1} = (\Phi_k + \epsilon_k \widehat{\Phi}_k) x_k + \zeta_k$. A great advantage of assumption (A1) is that it covers situations in which the signal evolution model is known, for both stationary and non-stationary signals (see, e.g., [23]). In addition, in uncertain systems with state-dependent multiplicative noise, as those considered in [6,32], the signal covariance function is factorizable, as it is shown in Section 5. Hence, assumption (A1) on the signal autocovariance function provides a unified context to deal with different situations based on the state-space model, avoiding the derivation of specific algorithms for each of them.*

2.2. Multisensor Observation Model

Assuming that there are m different sensors, the measured outputs before transmission, $z_k^{(i)} \in \mathbb{R}^{n_z}$, are described by the following observation model:

$$z_k^{(i)} = H_k^{(i)} x_k + v_k^{(i)}, \quad k \geq 1; \quad i = 1, \dots, m, \tag{1}$$

where the measurement matrices, $H_k^{(i)}$, and the noise vectors, $v_k^{(i)}$, satisfy the following assumptions:

(A2) $\{H_k^{(i)}; k \geq 1\}, i = 1, \dots, m$, are independent sequences of independent random parameter matrices, whose entries have known means and second-order moments; we will denote $\overline{H}_k^{(i)} \equiv E[H_k^{(i)}], k \geq 1$.

(A3) $\{v_k^{(i)}; k \geq 1\}, i = 1, \dots, m$, are white noise sequences with zero mean and known second-order moments, satisfying $E[v_k^{(i)} v_s^{(j)T}] = R_k^{(ij)} \delta_{k,s}, i, j = 1, \dots, m$.

Remark 2 (on assumption (A2)). *Usually, in network environments, the measurements are subject to different network-induced random phenomena and new estimation algorithms must be designed to incorporate the effects of these random uncertainties. For example, in systems with stochastic sensor gain degradation or missing measurements as those considered in [6,7], respectively, or in networked systems involving stochastic multiplicative noises in the state and measurement equations (see, e.g., [31,32]), new estimation algorithms are proposed since the conditions necessary to implement the conventional ones are not met. The aforementioned systems are particular cases of systems with random measurement matrices, and, hence, assumption (A2) allows for designing a unique estimation algorithm, which is suitable to address all of these situations involving random uncertainties. In addition, based on an augmentation approach, random measurement matrices can be used to model the measured outputs of sensor networks with random delays and packet dropouts (see, e.g., [8–10,20]). Therefore, assumption (A2) provides a unified framework to deal with a great variety of network-induced random phenomena as those mentioned above.*

2.3. Measurement Model with Transmission Random Delays and Packet Losses

Assuming that the maximum time delay is D , the measured output of the i -th sensor at time $r, z_r^{(i)}$, is transmitted during the sampling times $r, r + 1, \dots, r + D$, but, at each sampling time $k > D$, only one of the measurements $z_{k-D}^{(i)}, \dots, z_k^{(i)}$ is processed. Consequently, at any time $k > D$, the measurement processed can either arrive on time or be delayed by $d = 1, \dots, D$ sampling periods, while at any time $k \leq D$, the measurement processed can be delayed only by $d = 1, \dots, k - 1$ sampling periods, since only $z_1^{(i)}, \dots, z_k^{(i)}$ are available. Assuming, moreover, that the transmissions are perturbed by additive

noises, the measurements received at the processing centre, impaired by random delays and packet losses, can be described by the following model:

$$y_k^{(i)} = \sum_{d=0}^{(k-1)\wedge D} \gamma_{d,k}^{(i)} z_{k-d}^{(i)} + w_k^{(i)}, \quad k \geq 1, \tag{2}$$

where the following assumptions on the random variables modelling the delays, $\gamma_{d,k}^{(i)}$, and the transmission noise, $w_k^{(i)}$, are required:

- (A4) For each $d = 0, 1, \dots, D$, $\{\gamma_{d,k}^{(i)}; k > d\}$, $i = 1, \dots, m$, are independent sequences of independent Bernoulli random variables with $P[\gamma_{d,k}^{(i)} = 1] = \bar{\gamma}_{d,k}^{(i)}$ and $\sum_{d=0}^{(k-1)\wedge D} \gamma_{d,k}^{(i)} \leq 1$, $k \geq 1$.
- (A5) $\{w_k^{(i)}; k \geq 1\}$, $i = 1, \dots, m$, are white noise sequences with zero mean and known second-order moments, satisfying $E[w_k^{(i)} w_s^{(j)T}] = Q_k^{(ij)} \delta_{k,s}$, $i, j = 1, \dots, m$.

Remark 3 (on assumption (A4)). For $i = 1, \dots, m$, when $\gamma_{0,k}^{(i)} = 1$, the transmission of the i -th sensor is perfect and neither delay nor loss occurs at time k ; that is, with probability $\bar{\gamma}_{0,k}^{(i)}$ the k -th measurement of the i -th sensor is received and processed on time. Since $\sum_{d=0}^{(k-1)\wedge D} \gamma_{d,k}^{(i)} \leq 1$, if $\gamma_{0,k}^{(i)} = 0$, there then exists at most one $d = 1, \dots, D$, such that $\gamma_{d,k}^{(i)} = 1$. If there exists d such that $\gamma_{d,k}^{(i)} = 1$ (which occurs with probability $\bar{\gamma}_{d,k}^{(i)}$), then the measurement is delayed by d sampling periods. Otherwise, $\gamma_{d,k}^{(i)} = 0$ for all d , meaning that the measurement gets lost during the transmission at time k with probability $1 - \sum_{d=0}^{(k-1)\wedge D} \bar{\gamma}_{d,k}^{(i)}$.

Finally, the following independence hypothesis is assumed:

- (A6) For $i = 1, \dots, m$ and $d = 0, 1, \dots, D$, the processes $\{x_k; k \geq 1\}$, $\{H_k^{(i)}; k \geq 1\}$, $\{v_k^{(i)}; k \geq 1\}$, $\{w_k^{(i)}; k \geq 1\}$ and $\{\gamma_{d,k}^{(i)}; k > d\}$ are mutually independent.

3. Problem Statement

Given the observation Equations (1) and (2) with random measurement matrices and transmission random delays and packet dropouts, our purpose is to find the LS linear estimator, $\hat{x}_{k/L}$, of the signal x_k based on the observations from the different sensors $\{y_1^{(i)}, \dots, y_L^{(i)}, i = 1, \dots, m\}$. Specifically, our aim is to obtain recursive algorithms for the predictor ($L < k$), filter ($L = k$) and fixed-point smoother (k fixed and $L > k$).

3.1. Stacked Observation Model

Since the measurements coming from the different sensors are all gathered and jointly processed at each sampling time k , we will consider the vector constituted by the measurements from all sensors, $y_k = (y_k^{(1)T}, \dots, y_k^{(m)T})^T$. More specifically, the observation Equations (1) and (2) of all sensors are combined, yielding the following stacked observation model:

$$\begin{aligned} z_k &= H_k x_k + v_k, \quad k \geq 1, \\ y_k &= \sum_{d=0}^{(k-1)\wedge D} \Gamma_{d,k} z_{k-d} + w_k, \quad k \geq 1, \end{aligned} \tag{3}$$

where $z_k = (z_k^{(1)T}, \dots, z_k^{(m)T})^T$, $H_k = (H_k^{(1)T}, \dots, H_k^{(m)T})^T$, $v_k = (v_k^{(1)T}, \dots, v_k^{(m)T})^T$, $w_k = (w_k^{(1)T}, \dots, w_k^{(m)T})^T$ and $\Gamma_{d,k} = \text{Diag}(\gamma_{d,k}^{(1)}, \dots, \gamma_{d,k}^{(m)}) \otimes I_{n_2}$.

Hence, the problem is to obtain the LS linear estimator of the signal, x_k , based on the measurements $\{y_1, \dots, y_L\}$, given in the observation Equation (3). Next, we present the statistical properties of the processes involved in Equation (3), from which the LS estimation algorithms of the signal will be derived; these properties are easily inferred from the assumptions (A1)–(A6).

(P1) $\{H_k; k \geq 1\}$ is a sequence of independent random parameter matrices with known means, $\bar{H}_k \equiv E[H_k] = (\bar{H}_k^{(1)T}, \dots, \bar{H}_k^{(m)T})^T$, and

$$E[H_k A_k A_k^T H_s^T] = E[H_k A_k B_s^T H_s^T] = (E[H_k^{(i)} A_k B_s^T H_s^{(j)T}])_{i,j=1,\dots,m}, \quad s \leq k,$$

where $E[H_k^{(i)} A_k B_s^T H_s^{(j)T}] = \bar{H}_k^{(i)} A_k B_s^T \bar{H}_s^{(j)T}$, for $j \neq i$ or $s \neq k$, and the entries of $E[H_k^{(i)} A_k B_k^T H_k^{(i)T}]$ are computed as follows:

$$(E[H_k^{(i)} A_k B_k^T H_k^{(i)T}])_{pq} = \sum_{a=1}^{n_x} \sum_{b=1}^{n_x} E[h_{pa}^{(i)}(k) h_{qb}^{(i)}(k)] (A_k B_k^T)_{ab}, \quad p, q = 1, \dots, n_z,$$

where $h_{pq}^{(i)}(k)$ denotes the (p, q) -entry of the matrix $H_k^{(i)}$.

(P2) The noises $\{v_k; k \geq 1\}$ and $\{w_k; k \geq 1\}$ are zero-mean sequences with known second-order moments given by the matrices $R_k \equiv (R_k^{(ij)})_{i,j=1,\dots,m}$ and $Q_k \equiv (Q_k^{(ij)})_{i,j=1,\dots,m}$.

(P3) $\{\Gamma_{d,k}; k > d\}$, $d = 0, 1, \dots, D$, are sequences of independent random matrices with known means, $\bar{\Gamma}_{d,k} \equiv E[\Gamma_{d,k}] = \text{Diag}(\bar{\gamma}_{d,k}^{(1)}, \dots, \bar{\gamma}_{d,k}^{(m)}) \otimes I_{n_z}$, and if we denote $\gamma_{d,k} = (\gamma_{d,k}^{(1)}, \dots, \gamma_{d,k}^{(m)})^T \otimes \mathbf{1}_{n_z}$ and $\bar{\gamma}_{d,k} = E[\gamma_{d,k}]$, the covariance matrices $\Sigma_{d,d'}^{\gamma_k} \equiv E[(\gamma_{d,k} - \bar{\gamma}_{d,k})(\gamma_{d',k} - \bar{\gamma}_{d',k})^T]$, for $d, d' = 0, 1, \dots, D$, are also known matrices. Specifically,

$$\Sigma_{d,d'}^{\gamma_k} = \text{Diag}(\text{Cov}(\gamma_{d,k}^{(1)}, \gamma_{d',k}^{(1)}), \dots, \text{Cov}(\gamma_{d,k}^{(m)}, \gamma_{d',k}^{(m)})) \otimes (\mathbf{1}_{n_z} \mathbf{1}_{n_z}^T), \quad (4)$$

$$\text{with } \text{Cov}(\gamma_{d,k}^{(i)}, \gamma_{d',k}^{(i)}) = \begin{cases} \bar{\gamma}_{d,k}^{(i)}(1 - \bar{\gamma}_{d,k}^{(i)}), & d' = d, \\ -\bar{\gamma}_{d,k}^{(i)} \bar{\gamma}_{d',k}^{(i)}, & d' \neq d. \end{cases}$$

Moreover, for any deterministic matrix S , the Hadamard product properties guarantee that

$$E[(\Gamma_{d,k} - \bar{\Gamma}_{d,k}) S (\Gamma_{d',k} - \bar{\Gamma}_{d',k})] = \Sigma_{d,d'}^{\gamma_k} \circ S.$$

(P4) For $d = 0, 1, \dots, D$, the signal, $\{x_k; k \geq 1\}$, and the processes $\{H_k; k \geq 1\}$, $\{v_k; k \geq 1\}$, $\{w_k; k \geq 1\}$ and $\{\Gamma_{d,k}; k > d\}$ are mutually independent.

Remark 4 (on the observation covariance matrices). From the previous properties, it is clear that the observation process $\{z_k; k \geq 1\}$ is a zero-mean sequence whose covariance function, $\Sigma_{k,s}^z \equiv E[z_k z_s^T]$, is obtained by the following expression:

$$\Sigma_{k,s}^z = E[H_k A_k B_s^T H_s^T] + R_k \delta_{k,s}, \quad s \leq k, \quad (5)$$

where $E[H_k A_k B_s^T H_s^T]$ and R_k are calculated as it is indicated in properties (P1) and (P2), respectively.

3.2. Innovation Approach to the LS Linear Estimation Problem

The proposed covariance-based recursive algorithms for the LS linear prediction, filtering and fixed-point smoothing estimators will be derived by an innovation approach. This approach consists of transforming the observation process $\{y_k; k \geq 1\}$ into an equivalent one of orthogonal vectors called an innovation process, which will be denoted $\{\mu_k; k \geq 1\}$ and defined by $\mu_k = y_k - \hat{y}_{k/k-1}$, where $\hat{y}_{k/k-1}$ is the orthogonal projection of y_k onto the linear space spanned by $\{\mu_1, \dots, \mu_{k-1}\}$. Since both

processes span the same linear subspace, the LS linear estimator of any random vector α_k based on the observations $\{y_1, \dots, y_N\}$, denoted by $\hat{\alpha}_{k/N}$, is equal to that based on the innovations $\{\mu_1, \dots, \mu_N\}$, and, denoting $\Pi_h = E[\mu_h \mu_h^T]$, the following general expression for the LS linear estimators of α_k is obtained

$$\hat{\alpha}_{k/N} = \sum_{h=1}^N E[\alpha_k \mu_h^T] \Pi_h^{-1} \mu_h. \tag{6}$$

Hence, to obtain the signal estimators, it is necessary to find an explicit formula beforehand for the innovations and their covariance matrices.

Innovation μ_L and Covariance Matrix Π_L . Applying orthogonal projections in Equation (3), it is clear that the innovation μ_L is given by

$$\mu_L = y_L - \sum_{d=0}^{(L-1) \wedge D} \bar{\Gamma}_{d,L} \hat{z}_{L-d/L-1}, \quad L \geq 2; \quad \mu_1 = y_1, \tag{7}$$

so it is necessary to obtain the one-stage predictor $\hat{z}_{L/L-1}$ and the estimators $\hat{z}_{L-d/L-1}$, for $d = 1, \dots, (L-1) \wedge D$, of the observation process.

In order to obtain the covariance matrix Π_L , we use Equation (3) to express the innovations as:

$$\mu_L = \sum_{d=0}^{(L-1) \wedge D} \left((\Gamma_{d,L} - \bar{\Gamma}_{d,L}) z_{L-d} + \bar{\Gamma}_{d,L} (z_{L-d} - \hat{z}_{L-d/L-1}) \right) + w_L, \quad L \geq 2 \tag{8}$$

and, taking into account that

$$E \left[(\Gamma_{d,L} - \bar{\Gamma}_{d,L}) z_{L-d} (z_{L-d'} - \hat{z}_{L-d'/L-1})^T \bar{\Gamma}_{d',L} \right] = 0, \quad \forall d, d',$$

we have

$$\begin{aligned} \Pi_L &= \sum_{d,d'=0}^{(L-1) \wedge D} \left(\Sigma_{d,d'}^{\gamma_L} \circ \Sigma_{L-d,L-d'}^z + \bar{\Gamma}_{d,L} P_{L-d,L-d'/L-1}^z \bar{\Gamma}_{d',L} \right) + Q_L, \quad L \geq 2; \\ \Pi_1 &= (\Sigma_{0,0}^{\gamma_1} + \bar{\gamma}_{0,1} \bar{\gamma}_{0,1}^T) \circ \Sigma_{1,1}^z + Q_1, \end{aligned} \tag{9}$$

where the matrices $\Sigma_{d,d'}^{\gamma_L}$ and $\Sigma_{L-d,L-d'}^z$ are given in the Equations (4) and (5), respectively, and $P_{L-d,L-d'/L-1}^z \equiv E[(z_{L-d} - \hat{z}_{L-d/L-1})(z_{L-d'} - \hat{z}_{L-d'/L-1})^T]$.

4. Least-Squares Linear Signal Estimators

In this section, we derive recursive algorithms for the LS linear estimators, $\hat{x}_{k/L}$, $k \geq 1$, of the signal x_k based on the observations $\{y_1, \dots, y_L\}$ given in Equation (3); namely, a prediction and filtering algorithm ($L \leq k$) and a fixed-point smoothing algorithm (k fixed and $L > k$) are designed.

4.1. Signal Predictor and Filter $\hat{x}_{k/L}$, $L \leq k$

From the general expression given in Equation (6), to obtain the LS linear estimators $\hat{x}_{k/L} = \sum_{h=1}^L E[x_k \mu_h^T] \Pi_h^{-1} \mu_h$, $L \leq k$, it is necessary to calculate the coefficients

$$\mathcal{X}_{k,h} \equiv E[x_k \mu_h^T] = E[x_k y_h^T] - E[x_k \hat{y}_{h/h-1}^T], \quad h \leq k.$$

- On the one hand, using Equation (3) together with the independence hypotheses and assumption (A1) on the signal covariance factorization, it is clear that

$$E[x_k y_h^T] = \sum_{d=0}^{(h-1) \wedge D} E[x_k (H_{h-d} x_{h-d} + v_{h-d})^T] \bar{\Gamma}_{d,h} = A_k \sum_{d=0}^{(h-1) \wedge D} B_{h-d}^T \bar{H}_{h-d}^T \bar{\Gamma}_{d,h}, \quad h \leq k.$$

- On the other hand, since $\hat{y}_{h/h-1} = \sum_{d=0}^{(h-1)\wedge D} \bar{\Gamma}_{d,h} \hat{z}_{h-d/h-1}$, $h \geq 2$, and taking into account that, from Equation (6), $\hat{z}_{h-d/h-1} = \sum_{j=1}^{h-1} \mathcal{Z}_{h-d,j} \Pi_j^{-1} \mu_j$ with $\mathcal{Z}_{h-d,j} = E[z_{h-d} \mu_j^T]$, the following identity holds

$$E[x_k \hat{y}_{h/h-1}^T] = \sum_{d=0}^{(h-1)\wedge D} \left(\sum_{j=1}^{h-1} \mathcal{X}_{k,j} \Pi_j^{-1} \mathcal{Z}_{h-d,j}^T \right) \bar{\Gamma}_{d,h}.$$

Therefore, it is easy to check that $\mathcal{X}_{k,h} = A_k \mathcal{E}_h$, $1 \leq h \leq k$, where \mathcal{E}_h is a function satisfying that

$$\begin{aligned} \mathcal{E}_h &= \sum_{d=0}^{(h-1)\wedge D} B_{h-d}^T \bar{H}_{h-d}^T \bar{\Gamma}_{d,h} - \sum_{d=0}^{(h-1)\wedge D} \left(\sum_{j=1}^{h-1} \mathcal{E}_j \Pi_j^{-1} \mathcal{Z}_{h-d,j}^T \right) \bar{\Gamma}_{d,h}, \quad h \geq 2, \\ \mathcal{E}_1 &= B_1^T \bar{H}_1^T \bar{\Gamma}_{0,1}. \end{aligned} \tag{10}$$

Hence, it is clear that the signal prediction and filtering estimators can be expressed as

$$\hat{x}_{k/L} = A_k e_L, \quad L \leq k, \quad k \geq 1, \tag{11}$$

where the vectors e_L are defined by $e_L = \sum_{h=1}^L \mathcal{E}_h \Pi_h^{-1} \mu_h$, for $L \geq 1$, with $e_0 = 0$, thus obeying the recursive relation

$$e_L = e_{L-1} + \mathcal{E}_L \Pi_L^{-1} \mu_L, \quad L \geq 1; \quad e_0 = 0. \tag{12}$$

Matrices \mathcal{E}_L . Taking into account the above relation, an expression for \mathcal{E}_L , $L \geq 1$, must be derived. For this purpose, Equation (10) is rewritten for $h = L$ as

$$\mathcal{E}_L = \sum_{d=0}^{(L-1)\wedge D} B_{L-d}^T \bar{H}_{L-d}^T \bar{\Gamma}_{d,L} - \sum_{d=0}^{(L-1)\wedge D} \left(\sum_{j=1}^{L-1} \mathcal{E}_j \Pi_j^{-1} \mathcal{Z}_{L-d,j}^T \right) \bar{\Gamma}_{d,L}, \quad L \geq 2,$$

and we examine the cases $d = 0$ and $d \geq 1$ separately:

- For $d = 0$, using Equation (3), it holds that $\mathcal{Z}_{L,j} = \bar{H}_L \mathcal{X}_{L,j} = \bar{H}_L A_L \mathcal{E}_j$, for $j < L$, and, by denoting $\Sigma_L^e = \sum_{h=1}^L \mathcal{E}_h \Pi_h^{-1} \mathcal{E}_h$, $L \geq 1$, we obtain that

$$\left(\sum_{j=1}^{L-1} \mathcal{E}_j \Pi_j^{-1} \mathcal{Z}_{L,j}^T \right) \bar{\Gamma}_{0,L} = \left(\sum_{j=1}^{L-1} \mathcal{E}_j \Pi_j^{-1} \mathcal{E}_j^T \right) A_L^T \bar{H}_L^T \bar{\Gamma}_{0,L} = \Sigma_{L-1}^e A_L^T \bar{H}_L^T \bar{\Gamma}_{0,L}.$$

- For $d \geq 1$, since $\mathcal{Z}_{L-d,j} = \bar{H}_{L-d} A_{L-d} \mathcal{E}_j$, for $j < L - d$, we can see that

$$\left(\sum_{j=1}^{L-1} \mathcal{E}_j \Pi_j^{-1} \mathcal{Z}_{L-d,j}^T \right) \bar{\Gamma}_{d,L} = \Sigma_{L-d-1}^e A_{L-d}^T \bar{H}_{L-d}^T \bar{\Gamma}_{d,L} + \left(\sum_{j=L-d}^{L-1} \mathcal{E}_j \Pi_j^{-1} \mathcal{Z}_{L-d,j}^T \right) \bar{\Gamma}_{d,L}.$$

By substituting the above sums in \mathcal{E}_L , it is deduced that

$$\begin{aligned} \mathcal{E}_L &= \sum_{d=0}^{(L-1)\wedge D} (B_{L-d} - A_{L-d} \Sigma_{L-d-1}^e)^T \bar{H}_{L-d}^T \bar{\Gamma}_{d,L} - \sum_{d=1}^{(L-1)\wedge D} \left(\sum_{j=L-d}^{L-1} \mathcal{E}_j \Pi_j^{-1} \mathcal{Z}_{L-d,j}^T \right) \bar{\Gamma}_{d,L}, \quad L \geq 2; \\ \mathcal{E}_1 &= B_1^T \bar{H}_1^T \bar{\Gamma}_{0,1}, \end{aligned} \tag{13}$$

where the matrices $\mathcal{Z}_{L-d,j}$, $j \geq L-d$, will be obtained in the next subsection, as they correspond to the observation smoothing estimators, and the matrices Σ_L^e are recursively obtained by

$$\Sigma_L^e = \Sigma_{L-1}^e + \mathcal{E}_L \Pi_L^{-1} \mathcal{E}_L^T, \quad L \geq 1; \quad \Sigma_0^e = 0. \tag{14}$$

Finally, from assumption (A1) and since the estimation errors are orthogonal to the estimators, we have that the error covariance matrices, $P_{k/L}^x \equiv E[(x_k - \hat{x}_{k/L})(x_k - \hat{x}_{k/L})^T]$, are given by

$$P_{k/L}^x = A_k (B_k - A_k \Sigma_L^e)^T, \quad L \leq k, \quad k \geq 1. \tag{15}$$

4.2. Estimators of the Observations $\hat{z}_{k/L}$, $k \geq 1$

As it has been already indicated, the Equation (7) require obtaining the observation estimators (predictor, filter and smoother). From the general expression for the estimators given in Equation (6), we have that $\hat{z}_{k/L} = \sum_{j=1}^L \mathcal{Z}_{k,j} \Pi_j^{-1} \mu_j$, with $\mathcal{Z}_{k,j} = E[z_k \mu_j^T]$. Next, recursive expressions will be derived separately for $L < k$ (predictors) and $L \geq k$ (filter and smoothers).

Observation Prediction Estimators. Since $\mathcal{Z}_{k,j} = \bar{H}_k A_k \mathcal{E}_j$, for $j < k$, we have that the prediction estimators of the observations are given by

$$\hat{z}_{k/L} = \bar{H}_k A_L e_L, \quad L < k, \quad k \geq 1. \tag{16}$$

Observation Filtering and Fixed-Point Smoothing Estimators. Clearly, the filter and fixed-point smoothers of the observations are obtained by the following recursive expression:

$$\hat{z}_{k/L} = \hat{z}_{k/L-1} + \mathcal{Z}_{k,L} \Pi_L^{-1} \mu_L, \quad L \geq k, \quad k \geq 1, \tag{17}$$

with initial condition given by the one-stage predictor $\hat{z}_{k/k-1} = \bar{H}_k A_k e_{k-1}$.

Hence, the matrices $\mathcal{Z}_{k,L}$ must be calculated for $L \geq k$. Since the innovation is a white process, $E[\hat{z}_{k/L-1} \mu_L^T] = 0$ and hence $\mathcal{Z}_{k,L} = E[z_k \mu_L^T] = E[(z_k - \hat{z}_{k/L-1}) \mu_L^T]$. Now using Equation (8) for μ_L and, taking into account that $E[(z_k - \hat{z}_{k/L-1}) z_{L-d}^T (\Gamma_{d,L} - \bar{\Gamma}_{d,L})] = 0, \quad \forall d$, we have

$$\mathcal{Z}_{k,L} = \sum_{d=0}^{(L-1) \wedge D} P_{k,L-d/L-1}^z \bar{\Gamma}_{d,L}, \quad L \geq k, \tag{18}$$

where $P_{k,L-d/L-1}^z \equiv E[(z_k - \hat{z}_{k/L-1})(z_{L-d} - \hat{z}_{L-d/L-1})^T]$.

Consequently, the error covariance matrices $P_{k,h/m}^z = E[(z_k - \hat{z}_{k/m})(z_h - \hat{z}_{h/m})^T]$ must be derived, for which the following two cases are analyzed separately:

- * For $m \geq k \wedge h$, using Equation (17) and taking into account that $\mathcal{Z}_{k,m} = E[(z_k - \hat{z}_{k/m-1}) \mu_m^T]$, it is easy to see that

$$P_{k,h/m}^z = P_{k,h/m-1}^z - \mathcal{Z}_{k,m} \Pi_m^{-1} \mathcal{Z}_{h,m}^T, \quad m \geq k \wedge h. \tag{19}$$

- * For $m < h \leq k$, using Equation (16), assumption (A1) and the orthogonality between the estimation errors and the estimators, we obtain

$$P_{k,h/m}^z = A_k B_h^T - \bar{H}_k A_k \Sigma_m^e A_h^T \bar{H}_h^T, \quad m < h \leq k. \tag{20}$$

4.3. Signal Fixed-Point Smoother $\hat{x}_{k/L}$, $L > k$

Starting with the filter, $\hat{x}_{k/k}$, and the filtering error covariance matrix, $P_{k/k}^x$, it is clear that the signal fixed-point smoother $\hat{x}_{k/L}$, $L > k$, and the corresponding error covariance matrix, $P_{k/L}^x \equiv E[(x_k - \hat{x}_{k/L})(x_k - \hat{x}_{k/L})^T]$, are obtained by

$$\begin{aligned} \hat{x}_{k/L} &= \hat{x}_{k/L-1} + \mathcal{X}_{k,L} \Pi_L^{-1} \mu_L, \quad L > k, \quad k \geq 1, \\ P_{k/L}^x &= P_{k/L-1}^x - \mathcal{X}_{k,L} \Pi_L^{-1} \mathcal{X}_{k,L}^T, \quad L > k, \quad k \geq 1. \end{aligned} \tag{21}$$

An analogous reasoning to that of Equation (18) leads to the following expression for the matrices $\mathcal{X}_{k,L}$:

$$\mathcal{X}_{k,L} = \sum_{d=0}^{(L-1) \wedge D} P_{k,L-d/L-1}^{xz} \bar{\Gamma}_{d,L}, \quad L > k, \tag{22}$$

where $P_{k,L-d/L-1}^{xz} \equiv E[(x_k - \hat{x}_{k/L-1})(z_{L-d} - \hat{z}_{L-d/L-1})^T]$.

The derivation of the error cross-covariance matrices $P_{k,h/m}^{xz} = E[(x_k - \hat{x}_{k/m})(z_h - \hat{z}_{h/m})^T]$ is similar to that of the matrices $P_{k,h/m}^z$, and they are given by

$$\begin{aligned} P_{k,h/m}^{xz} &= P_{k,h/m-1}^{xz} - \mathcal{X}_{k,m} \Pi_m^{-1} \mathcal{Z}_{h,m}^T, & m \geq k \wedge h, \\ P_{k,h/m}^{xz} &= A_k (B_h - A_h \Sigma_m^e)^T H_h^T, & m < h \leq k, \\ P_{k,h/m}^{xz} &= (B_k - A_k \Sigma_m^e) A_h^T H_h^T, & m < k \leq h, \end{aligned} \tag{23}$$

where $\mathcal{X}_{k,m} = A_k \mathcal{E}_m$, for $m \leq k$, and $\mathcal{Z}_{h,m} = \bar{H}_h A_m \mathcal{E}_m$, for $m < h$, and otherwise these matrices are given by Equations (22) and (18), respectively.

4.4. Recursive Algorithms: Computational Procedure

The computational procedure of the proposed prediction, filtering and fixed-point smoothing algorithms can be summarized as follows:

- (1) *Covariance Matrices.* The covariance matrices $\Sigma_{d,d'}^{\gamma_k}$ and Σ_k^z are obtained by Equations (4) and (5), respectively; these matrices only depend on the system model information, so they can be calculated offline before the observed data packets are available.
- (2) *LS Linear Prediction and Filtering Recursive Algorithm.* At the sampling time k , once the $(k - 1)$ -th iteration is finished and \mathcal{E}_{k-1} , Π_{k-1} , Σ_{k-1}^e and e_{k-1} are known, the proposed prediction and filtering algorithm operates as follows:
 - (2a) Compute $\mathcal{Z}_{k,k-1} = \bar{H}_k A_k \mathcal{E}_{k-1}$ and $\mathcal{Z}_{k-d,k-1}$, for $d = 1, \dots, (k - 1) \wedge D$, by Equation (18). From these matrices, we obtain the observation estimators $\hat{z}_{k-d/k-1}$, for $d = 0, 1, \dots, (k - 1) \wedge D$, by Equation (19) and (20), and the observation error covariance matrices $P_{k-d,k-d'/k-1}^z$, for $d, d' = 0, 1, \dots, (k - 1) \wedge D$, by Equation (19) and (20).
 - (2b) Compute \mathcal{E}_k by Equation (13) and use $P_{k-d,k-d'/k-1}^z$ to obtain the innovation covariance matrix Π_k by Equation (9). Then, Σ_k^e is obtained by Equation (14) and, from it, the prediction and filtering error covariance matrices, $P_{k/k-s}^x$ and $P_{k/k}^x$, respectively, are obtained by Equation (15).
 - (2c) When the new measurement y_k is available, the innovation μ_k is computed by Equation (7) using $\hat{z}_{k-d/k-1}$, for $d = 0, 1, \dots, (k - 1) \wedge D$, and, from the innovation, e_k is obtained by Equation (12). Then, the predictors, $\hat{x}_{k/k-s}$ and the filter, $\hat{x}_{k/k}$ are computed by Equation (11).
- (3) *LS linear fixed-point smoothing recursive algorithm.* Once the filter, $\hat{x}_{k/k}$, and the filtering error covariance matrix, $P_{k/k}^x$ are available, the proposed smoothing estimators and the corresponding error covariance matrix are obtained as follows:

For $L = k + 1, k + 2, \dots$, compute the error cross-covariance matrices $P_{k,L-d/L-1}^{xz}$, for $d = 0, 1, \dots, (k - 1) \wedge D$, using Equation (23) and, from these matrices, $\mathcal{X}_{k,L}$ is derived by Equation (22); then, the smoothers $\hat{x}_{k/L}$ and their error covariance matrices $P_{k/L}^x$ are obtained from Equation (21).

5. Computer Simulation Results

In this section, a numerical example is presented with the following purposes: (a) to show that, although the current covariance-based estimation algorithms do not require the evolution model generating the signal process, they are also applicable to the conventional formulation using the state-space model, even in the presence of state-dependent multiplicative noise; (b) to illustrate some kinds of uncertainties which can be covered by the current model with random measurement matrices; and (c) to analyze how the estimation accuracy of the proposed algorithms is influenced by the sensor uncertainties and the random delays in the transmissions.

Signal Evolution Model with State-Dependent Multiplicative Noise. Consider a scalar signal $\{x_k; k \geq 0\}$ whose evolution is given by the following model with multiplicative and additive noises:

$$x_k = (0.9 + 0.01\epsilon_{k-1})x_{k-1} + \zeta_{k-1}, \quad k \geq 1,$$

where x_0 is a standard Gaussian variable and $\{\epsilon_k; k \geq 0\}$, $\{\zeta_k; k \geq 0\}$ are zero-mean Gaussian white processes with unit variance. Assuming that x_0 , $\{\epsilon_k; k \geq 0\}$ and $\{\zeta_k; k \geq 0\}$ are mutually independent, the signal covariance function is given by $E[x_k x_s] = 0.9^{k-s} D_s$, $s \leq k$, where $D_s = E[x_s^2]$ is recursively obtained by $D_s = 0.8101 D_{s-1} + 1$, for $s \geq 1$, with $D_0 = 1$; hence, the signal process satisfies assumption (A1) taking, for example, $A_k = 0.9^k$ and $B_s = 0.9^{-s} D_s$.

Sensor Measured Outputs. As in [22], let us consider scalar measurements provided by four sensors with different types of uncertainty: continuous and discrete gain degradation in sensors 1 and 2, respectively, missing measurements in sensor 3, and both missing measurements and multiplicative noise in sensor 4. These uncertainties can be described in a unified way by the current model with random measurement matrices; specifically, the measured outputs are described according to Equation (1):

$$z_k^{(i)} = H_k^{(i)} x_k + v_k^{(i)}, \quad k \geq 1, \quad i = 1, 2, 3, 4,$$

with the following characteristics:

- For $i = 1, 2, 3$, $H_k^{(i)} = C^{(i)} \theta_k^{(i)}$ and $H_k^{(4)} = (C^{(4)} + C^{(4')}\rho_k)\theta_k^{(4)}$, where $C^{(1)} = C^{(3)} = 0.8$, $C^{(2)} = C^{(4)} = 0.75$, $C^{(4')} = 0.95$, and $\{\rho_k; k \geq 1\}$ is a zero-mean Gaussian white process with unit variance. The sequences $\{\rho_k; k \geq 1\}$ and $\{\theta_k^{(i)}; k \geq 1\}$, $i = 1, 2, 3, 4$, are mutually independent, and $\{\theta_k^{(i)}; k \geq 1\}$, $i = 1, 2, 3, 4$, are white processes with the following time-invariant probability distributions:
 - $\theta_k^{(1)}$ is uniformly distributed over the interval $[0.1, 0.9]$;
 - $P[\theta_k^{(2)} = 0] = 0.3$, $P[\theta_k^{(2)} = 0.5] = 0.3$ $P[\theta_k^{(2)} = 1] = 0.4$;
 - For $i = 3, 4$, $\theta_k^{(i)}$ are Bernoulli random variables with the same time-invariant probabilities in both sensors $P[\theta_k^{(i)} = 1] = p$.
- The additive noises are defined by $v_k^{(i)} = c_i \eta_k^i$, $i = 1, 2, 3, 4$, where $c_1 = 0.5$, $c_2 = c_3 = 0.75$, $c_4 = 1$, and $\{\eta_k^i; k \geq 1\}$ is a zero-mean Gaussian white process with unit variance. Clearly, the additive noises $\{v_k^{(i)}; k \geq 1\}$, $i = 1, 2, 3, 4$, are correlated at any time, with $R_k^{(ij)} = c_i c_j$, $i, j = 1, 2, 3, 4$.

Observations with Bounded Random Delays and Packet Dropouts. Next, according to the theoretical observation model, let us suppose that bounded random measurement delays and packet dropouts,

with different delay probabilities, exist in the data transmission. Specifically, assuming that the largest delay is $D = 3$, let us consider the observation Equation (2):

$$y_k^{(i)} = \sum_{d=0}^{(k-1) \wedge 3} \gamma_{d,k}^{(i)} z_{k-d}^{(i)} + w_k^{(i)}, \quad k \geq 1,$$

where, for $i = 1, 2, 3, 4$ and $d = 0, 1, 2, 3$, $\{\gamma_{d,k}^{(i)}; k > d\}$, are sequences of independent Bernoulli variables with the same time-invariant delay probabilities for the four sensors $\bar{\gamma}_{d,k}^{(i)} = \bar{\gamma}_d$, where $\sum_{d=0}^3 \bar{\gamma}_d \leq 1$.

Hence, the packet dropout probability is $1 - \sum_{d=0}^3 \bar{\gamma}_d$. The transmission noise is defined by $w_k^{(i)} = c_i \eta_k^w$, $i = 1, 2, 3, 4$, where $\{\eta_k^w; k \geq 1\}$ is a zero-mean Gaussian white process with unit variance.

Finally, in order to apply the proposed algorithms, and according to (A5), we will assume that all of the processes involved in the observation equations are mutually independent.

To illustrate the feasibility and effectiveness of the proposed algorithms, they were implemented in MATLAB (R2011b 7.13.0.564, The Mathworks, Natick, MA, USA) and one hundred iterations of the prediction, filtering and fixed-point smoothing algorithms have been performed. In order to analyze the effect of the network-induced uncertainties on the estimation accuracy, different values of the probabilities p of the Bernoulli random variables that model the uncertainties of the third and fourth sensors, and several values of the delay probabilities $\bar{\gamma}_d$, $d = 0, 1, 2, 3$, have been considered.

Performance of the Prediction, Filtering and Fixed-Point Smoothing Estimators. Considering the values $p = 0.5$, $\bar{\gamma}_0 = 0.6$ (packet arrival probability) and $\bar{\gamma}_d = \frac{1-\bar{\gamma}_0}{4}$, $d = 1, 2, 3$ (delay probabilities), Figure 1 displays a simulated trajectory along with the prediction, filtering and smoothing estimations, showing a satisfactory and efficient tracking performance of the proposed estimators. Figure 2 shows the error variances of the predictors $\hat{z}_{k/k-2}$ and $\hat{z}_{k/k-1}$, the filter $\hat{z}_{k/k}$ and the smoothers $\hat{z}_{k/k+1}$ and $\hat{z}_{k/k+2}$. Analogously to what happens for non-delayed observations, the performance of the estimators becomes better as more observations are used; that is, the error variances of the smoothing estimators are less than the filtering ones which, in turn, are less than those of the predictors. Hence, the estimation accuracy of the smoothers is superior to that of the filter and predictors and improves as the number of iterations in the fixed-point smoothing algorithm increases. The performance of the proposed filter has also been evaluated in comparison with the standard Kalman filter; for this purpose, the filtering mean square error (MSE) at each sampling time was calculated by considering one thousand independent simulations and one hundred iterations of each filter. The results of this comparison are displayed in Figure 3, which shows that the proposed filter performs better than the Kalman filter, a fact that was expected since the latter does not take into account either the uncertainties in the measured outputs or the delays and losses during transmission.

Influence of the Missing Measurements. To analyze the sensitivity of the estimation performance on the effect of the missing measurements phenomenon in the third and fourth sensors, the error variances are calculated for different values of the probability p . Specifically, considering again $\bar{\gamma}_0 = 0.6$ and $\bar{\gamma}_d = 0.1$, $d = 1, 2, 3$, Figure 4 displays the prediction, filtering and fixed-point smoothing error variances for the values $p = 0.5$ to $p = 0.9$. This Figure shows that, as p increases, the estimation error variances become smaller and, hence, as it was expected, the performance of the estimators improves as the probability of missing measurements $1 - p$ decreases.

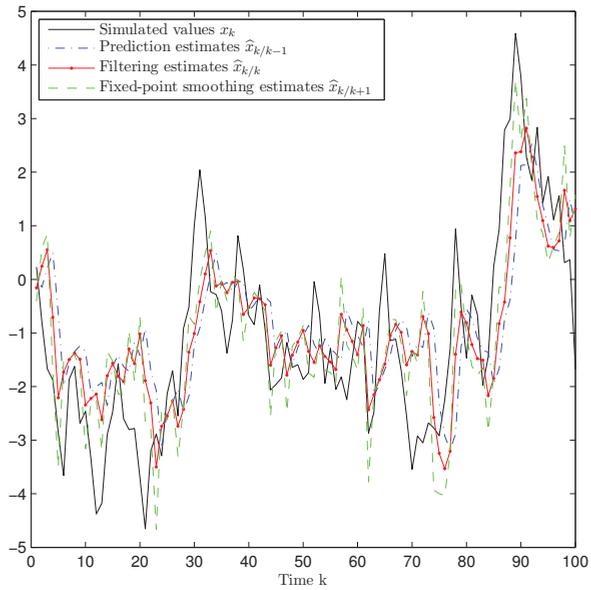


Figure 1. Simulated signal and proposed prediction, filtering and smoothing estimates when $p = 0.5$ and $\bar{\gamma}_0 = 0.6, \bar{\gamma}_d = 0.1, d = 1, 2, 3$.

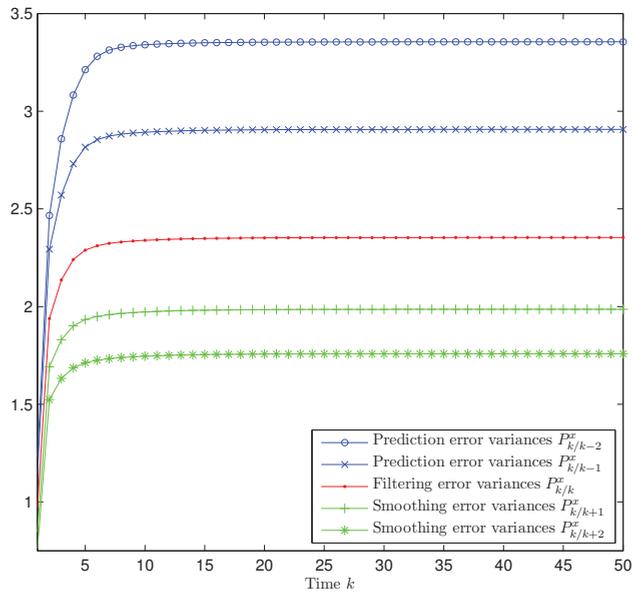


Figure 2. Prediction, filtering and smoothing error variances when $p = 0.5$ and $\bar{\gamma}_0 = 0.6, \bar{\gamma}_d = 0.1, d = 1, 2, 3$.

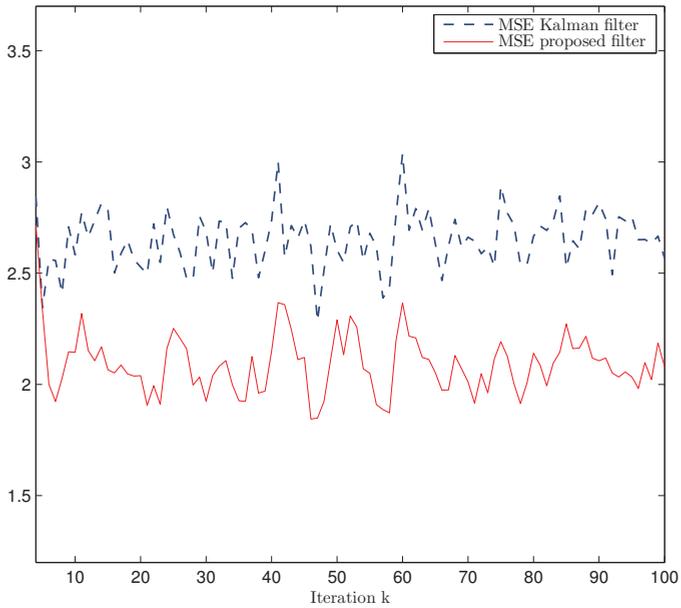


Figure 3. MSE of Kalman filter and proposed filter when $p = 0.5$ and $\bar{\gamma}_0 = 0.6, \bar{\gamma}_d = 0.1, d = 1, 2, 3$.

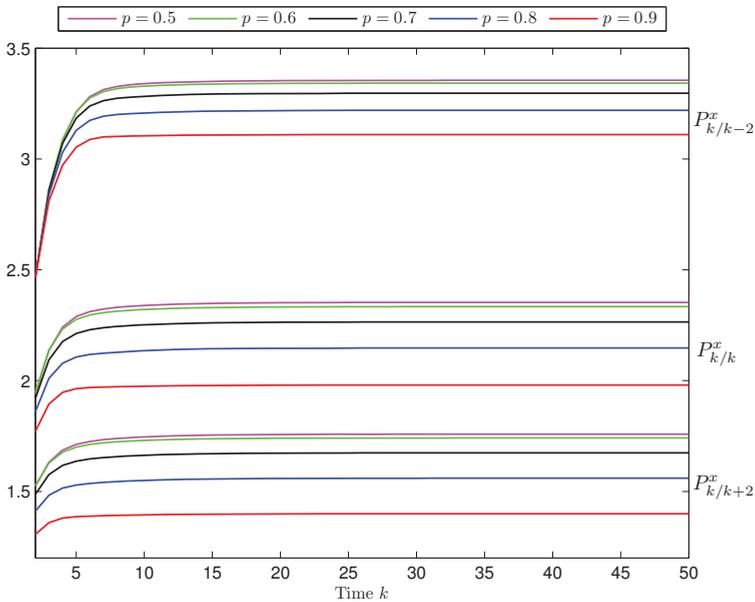


Figure 4. Prediction, filtering and fixed-point smoothing error variances for different values of the probability p , when $\bar{\gamma}_0 = 0.6$ and $\bar{\gamma}_d = 0.1, d = 1, 2, 3$.

Influence of the Transmission Random Delays and Packet Dropouts. Considering a fixed value of the probability p , namely, $p = 0.5$, in order to show how the estimation accuracy is influenced by

the transmission random delays and packet dropouts, the prediction, filtering and smoothing error variances are displayed in Figure 5 when $\bar{\gamma}_0$ is varied from 0.1 to 0.9, considering again $\bar{\gamma}_d = \frac{1-\bar{\gamma}_0}{4}$, $d = 1, 2, 3$. Since the behaviour of the error variances is analogous from a certain iteration on, only the results at the iteration $k = 50$ are shown in Figure 5. From this figure, we see that the performance of the estimators (predictor, filter and smoother) is indeed influenced by the transmission delay and packet dropout probabilities and, as it was expected, it is confirmed that the error variances become smaller, and hence the performance of the estimators improves, as the packet arrival probability $\bar{\gamma}_0$ increases. Moreover, for the filter and the smoothers, this improvement is more significant than for the predictors. In addition, as it was deduced from Figure 2, it is observed that, for all the values of $\bar{\gamma}_0$, the performance of the estimators is better as more observations are used, and this improvement is more significant as $\bar{\gamma}_0$ increases.

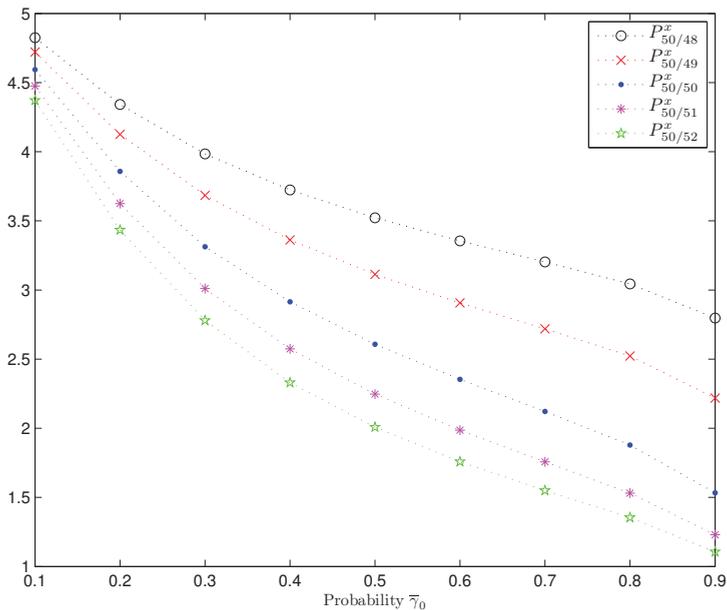


Figure 5. Estimation error variances for different values of the probability $\bar{\gamma}_0$, when $p = 0.5$.

Finally, it must be remarked that analogous conclusions are deduced for other values of the probabilities p and $\bar{\gamma}_d$, $d = 0, 1, 2, 3$, and also when such probabilities are different at the different sensors.

6. Conclusions

This paper makes valuable contributions to the optimal fusion estimation problem in networked stochastic systems with random parameter matrices, when multi-step delays or even packet dropouts occur randomly during the data transmission. By an innovation approach, recursive prediction, filtering and fixed-point smoothing algorithms have been designed, which are easily implementable and do not require the signal evolution model, but only the mean and covariance functions of the system processes.

Unlike other estimation algorithms proposed in the literature, where the estimators are restricted to obey a particular structure, in this paper, recursive optimal estimation algorithms are designed without requiring a particular structure on the estimators, but just using the LS optimality criterion.

Another advantage is that the current approach does not resort to the augmentation technique and, consequently, the dimension of the designed estimators is the same as that of the original signal, thus reducing the computational burden in the processing centre.

Acknowledgments: This research is supported by the “Ministerio de Economía y Competitividad” and “Fondo Europeo de Desarrollo Regional” FEDER (Grant No. MTM2014-52291-P).

Author Contributions: All of the authors contributed equally to this work. Raquel Caballero-Águila, Aurora Hermoso-Carazo and Josefa Linares-Pérez provided original ideas for the proposed model and collaborated in the derivation of the estimation algorithms; they participated equally in the design and analysis of the simulation results; and the paper was also written and reviewed cooperatively.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ran, C.; Deng, Z. Self-tuning weighted measurement fusion Kalman filtering algorithm. *Comput. Stat. Data Anal.* **2012**, *56*, 2112–2128.
2. Feng, J.; Zeng, M. Optimal distributed Kalman filtering fusion for a linear dynamic system with cross-correlated noises. *Int. J. Syst. Sci.* **2012**, *43*, 385–398.
3. Yan, L.; Li, X.; Xia, Y.; Fu, M. Optimal sequential and distributed fusion for state estimation in cross-correlated noise. *Automatica* **2013**, *49*, 3607–3612.
4. Ma, J.; Sun, S. Centralized fusion estimators for multisensor systems with random sensor delays, multiple packet dropouts and uncertain observations. *IEEE Sens. J.* **2013**, *13*, 1228–1235.
5. Gao, S.; Chen, P. Suboptimal filtering of networked discrete-time systems with random observation losses. *Math. Probl. Eng.* **2014**, *2014*, 151836.
6. Liu, Y.; He, X.; Wang, Z.; Zhou, D. Optimal filtering for networked systems with stochastic sensor gain degradation. *Automatica* **2014**, *50*, 1521–1525.
7. Chen, B.; Zhang, W.; Yu, L. Distributed fusion estimation with missing measurements, random transmission delays and packet dropouts. *IEEE Trans. Autom. Control* **2014**, *59*, 1961–1967.
8. Li, N.; Sun, S.; Ma, J. Multi-sensor distributed fusion filtering for networked systems with different delay and loss rates. *Digit. Signal Process.* **2014**, *34*, 29–38.
9. Wang, S.; Fang, H.; Tian, X. Recursive estimation for nonlinear stochastic systems with multi-step transmission delays, multiple packet dropouts and correlated noises. *Signal Process.* **2015**, *115*, 164–175.
10. Chen, D.; Yu, Y.; Xu, L.; Liu, X. Kalman filtering for discrete stochastic systems with multiplicative noises and random two-step sensor delays. *Discret. Dyn. Nat. Soc.* **2015**, *2015*, 809734.
11. García-Ligero, M.J.; Hermoso-Carazo, A.; Linares-Pérez, J. Distributed fusion estimation in networked systems with uncertain observations and markovian random delays. *Signal Process.* **2015**, *106*, 114–122.
12. Gao, S.; Chen, P.; Huang, D.; Niu, Q. Stability analysis of multi-sensor Kalman filtering over lossy networks. *Sensors* **2016**, *16*, 566.
13. Lin, H.; Sun, S. State estimation for a class of non-uniform sampling systems with missing measurements. *Sensors* **2016**, *16*, 1155.
14. Hu, J.; Wang, Z.; Chen, D.; Alsaadi, F.E. Estimation, filtering and fusion for networked systems with network-induced phenomena: New progress and prospects. *Inf. Fusion* **2016**, *31*, 65–75.
15. Sun, S.; Lin, H.; Ma, J.; Li, X. Multi-sensor distributed fusion estimation with applications in networked systems: A review paper. *Inf. Fusion* **2017**, *38*, 122–134.
16. Li, W.; Jia, Y.; Du, J. Distributed filtering for discrete-time linear systems with fading measurements and time-correlated noise. *Digit. Signal Process.* **2017**, *60*, 211–219.
17. Luo, Y.; Zhu, Y.; Luo, D.; Zhou, J.; Song, E.; Wang, D. Globally optimal multisensor distributed random parameter matrices Kalman filtering fusion with applications. *Sensors* **2008**, *8*, 8086–8103.
18. Shen, X.J.; Luo, Y.T.; Zhu, Y.M.; Song, E.B. Globally optimal distributed Kalman filtering fusion. *Sci. China Inf. Sci.* **2012**, *55*, 512–529.
19. Hu, J.; Wang, Z.; Gao, H. Recursive filtering with random parameter matrices, multiple fading measurements and correlated noises. *Automatica* **2013**, *49*, 3440–3448.

20. Linares-Pérez, J.; Caballero-Águila, R.; García-Garrido, I. Optimal linear filter design for systems with correlation in the measurement matrices and noises: Recursive algorithm and applications. *Int. J. Syst. Sci.* **2014**, *45*, 1548–1562.
21. Yang, Y.; Liang, Y.; Pan, Q.; Qin, Y.; Yang, F. Distributed fusion estimation with square-root array implementation for Markovian jump linear systems with random parameter matrices and cross-correlated noises. *Inf. Sci.* **2016**, *370–371*, 446–462.
22. Caballero-Águila, R.; Hermoso-Carazo, A.; Linares-Pérez, J. Networked fusion filtering from outputs with stochastic uncertainties and correlated random transmission delays. *Sensors* **2016**, *16*, 847.
23. Caballero-Águila, R.; Hermoso-Carazo, A.; Linares-Pérez, J. Fusion estimation using measured outputs with random parameter matrices subject to random delays and packet dropouts. *Signal Process.* **2016**, *127*, 12–23.
24. Feng, J.; Zeng, M. Descriptor recursive estimation for multiple sensors with different delay rates. *Int. J. Control* **2011**, *84*, 584–596.
25. Caballero-Águila, R.; Hermoso-Carazo, A.; Linares-Pérez, J. Least-squares linear-estimators using measurements transmitted by different sensors with packet dropouts. *Digit. Signal Process.* **2012**, *22*, 1118–1125.
26. Ma, J.; Sun, S. Distributed fusion filter for networked stochastic uncertain systems with transmission delays and packet dropouts. *Signal Process.* **2017**, *130*, 268–278.
27. Wang, S.; Fang, H.; Liu, X. Distributed state estimation for stochastic non-linear systems with random delays and packet dropouts. *IET Control Theory Appl.* **2015**, *9*, 2657–2665.
28. Sun, S. Linear minimum variance estimators for systems with bounded random measurement delays and packet dropouts. *Signal Process.* **2009**, *89*, 1457–1466.
29. Sun, S. Optimal linear filters for discrete-time systems with randomly delayed and lost measurements with/without time stamps. *IEEE Trans. Autom. Control* **2013**, *58*, 1551–1556.
30. Sun, S.; Ma, J. Linear estimation for networked control systems with random transmission delays and packet dropouts. *Inf. Sci.* **2014**, *269*, 349–365.
31. Wang, S.; Fang, H.; Tian, X. Minimum variance estimation for linear uncertain systems with one-step correlated noises and incomplete measurements. *Digit. Signal Process.* **2016**, *49*, 126–136.
32. Tian, T.; Sun, S.; Li, N. Multi-sensor information fusion estimators for stochastic uncertain systems with correlated noises. *Inf. Fusion* **2016**, *27*, 126–137.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Multisensor Parallel Largest Ellipsoid Distributed Data Fusion with Unknown Cross-Covariances

Baoyu Liu ¹, Xingqun Zhan ^{1,*} and Zheng H. Zhu ²

¹ School of Aeronautics and Astronautics, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang District, Shanghai 200240, China; abao-liu@163.com

² Department of Mechanical Engineering, York University, 4700 Keele Street, Toronto, ON M3J 1P3, Canada; gzhu@yorku.ca

* Correspondence: xqzhan@sjtu.edu.cn; Tel.: +86-21-3420-4306

Received: 26 April 2017; Accepted: 23 June 2017; Published: 29 June 2017

Abstract: As the largest ellipsoid (LE) data fusion algorithm can only be applied to two-sensor system, in this contribution, parallel fusion structure is proposed to introduce the LE algorithm into a multisensor system with unknown cross-covariances, and three parallel fusion structures based on different estimate pairing methods are presented and analyzed. In order to assess the influence of fusion structure on fusion performance, two fusion performance assessment parameters are defined as Fusion Distance and Fusion Index. Moreover, the formula for calculating the upper bounds of actual fused error covariances of the presented multisensor LE fusers is also provided. Demonstrated with simulation examples, the Fusion Index indicates fuser's actual fused accuracy and its sensitivity to the sensor orders, as well as its robustness to the accuracy of newly added sensors. Compared to the LE fuser with sequential structure, the LE fusers with proposed parallel structures not only significantly improve their properties in these aspects, but also embrace better performances in consistency and computation efficiency. The presented multisensor LE fusers generally have better accuracies than covariance intersection (CI) fusion algorithm and are consistent when the local estimates are weakly correlated.

Keywords: largest ellipsoid; distributed data fusion; parallel structure; unknown cross-covariances; multisensor

1. Introduction

Multiple sensors have been widely employed in various systems, such as the integrated navigation system of driverless cars. Multisensor data fusion aims to achieve an accurate, robust and reliable representative of the target of interest by combining the information from different used sensors. The data (estimate) fusion algorithms of multisensor system can be generally classified to centralized fusion algorithms and distributed fusion algorithms. The centralized fusion algorithms can obtain the globally optimal estimate by directly combining sensor outputs to an augmented measurement. However, such fusion architecture leads to a heavy computational burden; furthermore, the fused estimate will be easily corrupted if any sensor degenerates. The distributed fusion algorithms can reduce the computational burden and facilitate fault detection or isolation more conveniently through combining the local estimates from sensors by weighting matrices. In the distributed data fusion of multisensor system, once the cross-covariances among local estimates are known exactly, globally optimal or suboptimal estimates can be obtained by using optimal fusion algorithms, such as the two-sensor Bar-Shalom and Campo (BC) algorithm [1] or its version for multisensor systems [2] in the sense of maximum likelihood (ML), or the optimal distributed Kalman fuser weighted by matrices [3] in the sense of linear unbiased minimum variance (LUMV). However, in many applications, these cross-covariances are difficult to be computed accurately; one critical issue of multisensor data

fusion in the case is how to merge the local estimates efficiently to achieve a fused estimate that has high accuracy and simultaneously is consistent. With this aim, various distributed data fusion algorithms for multisensor systems with unknown cross-covariances are proposed, such as the convex combination (CC) algorithm [4], ellipsoidal intersection (EI) algorithm [5], largest ellipsoid (LE) algorithm [6], covariance intersection (CI) algorithm [7] and their variants the internal ellipsoidal approximation (IEA) algorithm [8], fast covariance intersection (FCI) algorithm [9], etc. The CI algorithm is of special concern and has been widely applied to many fields, for it yields a common upper bound of actual fused error covariance regardless of unknown cross-covariances. When all the local estimates are consistent, the CI algorithm gives a consistent fused estimate with higher accuracy than each local estimate. However, the CI algorithm is based on the optimization of a multi-dimensional nonlinear cost function, which results in a large computational burden; in addition, it overestimates the actual fused error covariance and pursues consistency at the expense of accuracy, which leads to a significant decrease in performance [8,10,11]. Although several improvements [12–17] have been developed for the CI algorithm since it was proposed, these drawbacks have not yet been essentially resolved. Compared to the CI algorithm, the CC algorithm and the EI algorithm, the LE algorithm does not need any optimizations of cost functions, but they may become inconsistent in some cases owing to the unknown cross-covariances.

The LE algorithm is a two-sensor fusion algorithm that obtains a new estimate from two local estimates based on a series of matrix transformations. Instead of computing a tightest fused error covariance ellipsoid which encloses the intersection region of the covariance ellipsoids of local estimates in the CI algorithm, the LE algorithm computes the largest ellipsoid contained within that intersection region, which leads to a tighter fused error covariance ellipsoid. Besides not requiring optimization of a cost function, the LE algorithm has many other advantages. It comparatively has better consistency performance than the CC algorithm and has better actual fused accuracy than the EI algorithm and CI algorithm in general. The two shortcomings of the LE algorithm are that the LE algorithm is limited to two-sensor applications and its consistency can not be unconditionally satisfied for correlated local estimates. However, although the cross-covariances are hard to be known exactly, some information about the dependency properties among local estimates might be possible to be obtained for users, such as the correlation level [11]. When the local estimates are weakly correlated, the adverse impact of the correlation on fusion consistency performance will be limited. On the other hand, although the performance of a fuser is basically determined by its fusion algorithm, the fusion structure also has an important influence on it. A sequential covariance intersection (SCI) Kalman filter is proposed by applying sequential processing to reduce the complexity and computational burden of the batch CI algorithm [18]. A two-level fusion structure is presented, which combines the merits of the measurement fusion algorithm and CI algorithm to reduce calculation burden and get a more accurate fused estimate [19]. In addition, Kalman-Particle filtering with a cascaded structure is conducted to reduce the complexity of a high dimensional state space model, which leads to an easier tuning and more precise debugging, as well as reduced computation time [20]. Therefore, when the local estimates are not strongly correlated, how to extend the LE algorithm to multisensor cases with proper fusion structure and simultaneously improve its performances in accuracy and consistency is worth being addressed.

This paper proposes a largest ellipsoid fusion Kalman filtering with parallel fusion structure for the data fusion of multisensor system with unknown cross-covariances among local estimates, which realizes the multisensor fusion as a tree form with each level consisting of one or a series of parallel LE fusions. With parallel fusion structure, the data processing task of the proposed filtering is amenable to multiprocessor implementation. Three different estimate pairing methods for constructing the parallel fusion structure are given, and two fusion performance assessment parameters of Fusion Distance and Fusion Index for assessing the influence of fusion structure on fusion performance are defined. The attributes of the presented fusers in Fusion Distance, Fusion Index, and accuracy relation based on covariance ellipsoid, as well as the formula for calculating the upper bounds of actual fused

error covariances of the presented fusers regardless of unknown cross-covariances, are provided. In order to verify the effectiveness of the proposed filtering, simulation examples are carried out to compare the performances of CI algorithm, optimal distributed Kalman fuser weighted by matrices, LE fuser with sequential structure and the LE fusers with the proposed parallel structures.

2. Preliminaries

An estimate of stochastic state $x \in R^{n \times 1}$ usually can be characterized with a Gaussian distribution $\hat{x}_e \sim N(\bar{x}_e, P_e)$, where $\bar{x}_e \in R^{n \times 1}$ and $P_e \in R^{n \times n}$, respectively, represent the mean and fused error covariance. The estimate is said to be consistent (or conservative) only when its actual fused error covariance $\bar{P}_e = E[(\hat{x}_e - x)(\hat{x}_e - x)^T]$ satisfies $\bar{P}_e \leq P_e$ [12,13], the superscript T denotes the transpose, and the notation $E(*)$ denotes the expectation. The fused error covariance represents the fused accuracy, and the actual fused error covariance indicates the actual fused accuracy. Given real symmetric positive definite matrices $P_a \in R^{n \times n}$ and $P_b \in R^{n \times n}$, $P_a \geq P_b$ denotes $P_a - P_b$ as positive semi-definite. Then, $\text{tr}(P_a) \geq \text{tr}(P_b)$, $P_a^{-1} \leq P_b^{-1}$ and $CP_aC^T \geq CP_bC^T$ hold for any row full rank matrix C [21], the superscript -1 denotes the inverse, and the notation $\text{tr}(*)$ denotes the trace. The estimate \hat{x}_e with Gaussian distribution also can be illustrated by multi-dimensional covariance ellipsoid whose contour of one sigma is defined by $\mathfrak{R}_{(\bar{x}_e, P_e)} \equiv \{x | (x - \bar{x}_e)^T P_e^{-1} (x - \bar{x}_e) = 1\}$. The center of $\mathfrak{R}_{(\bar{x}_e, P_e)}$ is \bar{x}_e , and the lengths of the semi-axes of $\mathfrak{R}_{(\bar{x}_e, P_e)}$ are given by $\sqrt{\sigma_i}$, where σ_i are the singular values of the matrix P_e . Hence, larger covariance ellipsoid means worse accuracy. For two estimates $\hat{x}_a \sim N(\bar{x}_a, P_a)$ and $\hat{x}_b \sim N(\bar{x}_b, P_b)$, the necessary and sufficient condition for $P_a \geq P_b$ is $\mathfrak{R}_{(\bar{x}_a, P_a)} \supset \mathfrak{R}_{(\bar{x}_b, P_b)}$, which means that the ellipsoid $\mathfrak{R}_{(\bar{x}_a, P_a)}$ encloses the ellipsoid $\mathfrak{R}_{(\bar{x}_b, P_b)}$.

Consider the discrete time-invariant linear stochastic system with multiple sensors

$$\begin{aligned} x(t+1) &= \Phi x(t) + \Gamma w(t) \\ y_i(t) &= H_i x(t) + v_i(t), \quad i = 1, 2, \dots, L \end{aligned} \tag{1}$$

where t is the discrete time, L is the number of sensors, $x(t) \in R^{n \times 1}$ is the state, $y_i(t) \in R^{m \times 1}$ is the measurement, $w(t)$ and $v_i(t) \in R^{m \times 1}$ are the uncorrelated white noises with zero mean and covariance matrices Q and R_i , respectively; Φ, Γ, H_i are constant matrices with compatible dimensions, and (Φ, Γ) is a completely controllable pair, (Φ, H_i) is a completely observable pair. The subsystem based on the i th sensor of multisensor system (1) has local steady-state Kalman filter as [22]

$$\hat{x}_i(t|t) = (I_n - K_i H_i) \Phi \hat{x}_i(t-1|t-1) + K_i y_i(t), \tag{2}$$

with $K_i = \Sigma_i H_i^T (H_i \Sigma_i H_i^T + R_i)^{-1}$, where I_n denotes the $n \times n$ unit matrix, Σ_i satisfies the Riccati equation

$$\Sigma_i = \Phi \left[\Sigma_i - \Sigma_i H_i^T (H_i \Sigma_i H_i^T + R_i)^{-1} H_i \Sigma_i \right] \Phi^T + \Gamma Q \Gamma^T. \tag{3}$$

The local filtering error covariance is given by

$$P_i = P_{ii} = (I_n - K_i H_i) \Sigma_i, \tag{4}$$

and the local filtering error cross-covariance between the subsystems of the i th and j th sensor satisfies the Lyapunov equation

$$P_{ij} = (I_n - K_i H_i) \Phi P_{ij} [(I_n - K_j H_j) \Phi]^T + (I_n - K_i H_i) \Gamma Q \Gamma^T (I_n - K_j H_j)^T, \tag{5}$$

Then, the overall error covariance of the multisensor system is $\Sigma = (P_{ij}) \in R^{Lm \times Ln}$, $i, j = 1, 2, \dots, L$.

3. Distributed Fusion Algorithms

For unbiased state estimation, both the state and its error covariance should be estimated.

3.1. Optimal Distributed Kalman Fuser Weighted by Matrices

Once all of the local filtering error covariances and cross-covariances are obtained, the optimal distributed Kalman fuser weighted by matrices under LUMV for multisensor system (1) is given by [3]

$$\hat{x}_{(L)}^O = \sum_{i=1}^L A_i^O \hat{x}_i(t|t), \tag{6}$$

where A_i^O is the optimal state estimation weighting matrix corresponding to the i th local estimate and computed by

$$[A_1^O, A_2^O, \dots, A_L^O] = (e^T \Sigma^{-1} e)^{-1} e^T \Sigma^{-1}, \tag{7}$$

where $e = [I_n, I_n, \dots, I_n]^T$ is a $Ln \times n$ matrix. The error covariance of $\hat{x}_{(L)}^O$ is given as

$$P_{(L)}^O = [A_1^O, A_2^O, \dots, A_L^O] \Sigma [A_1^O, A_2^O, \dots, A_L^O]^T = (e^T \Sigma^{-1} e)^{-1}, \tag{8}$$

with the accuracy relation $P_{(L)}^O \leq P_i, i = 1, 2, \dots, L$. As shown from (6)–(8), it is necessary that all the covariances $P_{ij}, i, j = 1, 2, \dots, L$ should be exactly known in the calculation of $\hat{x}_{(L)}^O$ and $P_{(L)}^O$; however, such a condition can not be satisfied in many applications. Moreover, Σ^{-1} is also required to be computed, which will result in heavy computational burden when the number of sensors is large.

3.2. Largest Ellipsoid Fusion Algorithm

The LE algorithm obtains a new estimate from two local estimates based on a series of matrix transformations. Given two local estimates $\hat{x}_1 \sim N(\bar{x}_1, P_1)$ and $\hat{x}_2 \sim N(\bar{x}_2, P_2)$. Firstly, we diagonalize P_1 as

$$U^T P_1 U = W = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), \tag{9}$$

where the notation $\text{diag}(\ast)$ denotes forming a diagonal matrix sequentially using the elements in parentheses, and U is an orthogonal matrix. Then, we execute the following transformations:

$$\begin{aligned} W^{-\frac{1}{2}} U^T P_1 U W^{-\frac{1}{2}} &= P'_1 = I_n \\ W^{-\frac{1}{2}} U^T P_2 U W^{-\frac{1}{2}} &= P'_2 \end{aligned} \tag{10}$$

where $W^{-\frac{1}{2}} = [\text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_n})]^{-1}$. Applying a second diagonalization, we have

$$\begin{aligned} V^T P'_1 V &= P_1^* = I_n \\ V^T P'_2 V &= P_2^* = \text{diag}(\lambda_1^*, \lambda_2^*, \dots, \lambda_n^*) \end{aligned} \tag{11}$$

where V is an orthogonal matrix. Then, we define

$$\begin{aligned} \hat{x}_1^* &= V^T W^{-\frac{1}{2}} U^T \hat{x}_1 \\ \hat{x}_2^* &= V^T W^{-\frac{1}{2}} U^T \hat{x}_2 \end{aligned} \tag{12}$$

After these transformations, we obtain two new estimates in the new Euclidean space R^* as \hat{x}_1^* and \hat{x}_2^* with error covariances P_1^* and P_2^* , respectively. From (11), we know that both P_1^* and P_2^* are diagonal matrices. The fused estimate of LE algorithm in Euclidean space R^* is given by

$$\hat{x}_{LE}^* = [(P_1^*)^{-1} + (P_2^*)^{-1}]^{-1} [(P_1^*)^{-1} \hat{x}_1^* + (P_2^*)^{-1} \hat{x}_2^*], \tag{13}$$

with fused error covariance

$$P_{LE}^* = SP_1^* + (I_n - S)P_2^*, \tag{14}$$

where $S = \text{diag}(s_1, s_2, \dots, s_n)$ with

$$s_i = \begin{cases} 1, & \lambda_i^* \geq 1 \\ 0, & \text{else} \end{cases}. \tag{15}$$

The fused estimate of LE algorithm in the original Euclidean space R is obtained by

$$\hat{x}_L^E = UW^{\frac{1}{2}}V\hat{x}_{LE}^* = A_1^{LE}\hat{x}_1 + A_2^{LE}\hat{x}_2, \tag{16}$$

with fused error covariance

$$P^{LE} = UW^{\frac{1}{2}}VP_{LE}^*V^TW^{\frac{1}{2}}U^T = B_1^{LE}P_1 + B_2^{LE}P_2, \tag{17}$$

where the weighting matrices are calculated as follows:

$$\begin{aligned} A_1^{LE} &= UW^{\frac{1}{2}}V[(P_1^*)^{-1} + (P_2^*)^{-1}]^{-1}(P_1^*)^{-1}V^TW^{-\frac{1}{2}}U^T \\ A_2^{LE} &= UW^{\frac{1}{2}}V[(P_1^*)^{-1} + (P_2^*)^{-1}]^{-1}(P_2^*)^{-1}V^TW^{-\frac{1}{2}}U^T \end{aligned} \tag{18}$$

$$\begin{aligned} B_1^{LE} &= UW^{\frac{1}{2}}VSV^TW^{-\frac{1}{2}}U^T \\ B_2^{LE} &= UW^{\frac{1}{2}}V(I_n - S)V^TW^{-\frac{1}{2}}U^T \end{aligned} \tag{19}$$

From (16)–(19), we know that both the fused state and fused error covariance of LE algorithm are the linear estimates. The covariance ellipsoid of P^{LE} is the largest one contained within the intersection region $\mathfrak{R}_{(\bar{x}_1, P_1)} \cap \mathfrak{R}_{(\bar{x}_2, P_2)}$, and it is obvious that $P^{LE} \leq P_1, P^{LE} \leq P_2$. For any two unbiased local estimates of state x with Gaussian distributions, because both the optimal distributed Kalman fuser estimate $\hat{x}_{(2)}^O$ and the LE algorithm estimate \hat{x}_L^E are linear unbiased estimates of x , and $\hat{x}_{(2)}^O$ is the LUMV estimate, we have $P_{(2)}^O \leq \bar{P}^{LE}$, where $\bar{P}^{LE} = E[(\hat{x}^{LE} - x)(\hat{x}^{LE} - x)^T]$ is the actual fused error covariance of the LE algorithm. Furthermore, it can be easily proven that $\bar{P}^{LE} < P^{LE}$ when P_1 and P_2 are independent.

There are mainly two drawbacks for the LE algorithm. The first one is that the LE algorithm can only handle two sensors at a time. The second one is that the LE algorithm can't guarantee its consistency, which implies that $\bar{P}^{LE} \leq P^{LE}$ will be unsatisfied in some cases. In the two-sensor case, according to [23], for any point $x \in \mathfrak{R}_{(\bar{x}_1, P_1)} \cap \mathfrak{R}_{(\bar{x}_2, P_2)}$, there is a feasible cross-covariance P_{12} that lets $x \in P_{(2)}^O$. As mentioned above, the covariance ellipsoid of P^{LE} is the largest one contained within the intersection region $\mathfrak{R}_{(\bar{x}_1, P_1)} \cap \mathfrak{R}_{(\bar{x}_2, P_2)}$, but it generally doesn't cover the whole intersection region. Combining the accuracy relation between the optimal distributed Kalman fuser and LE algorithm, if the P_{12} lets $P_{(2)}^O \not\leq P^{LE}$, $\bar{P}^{LE} \leq P^{LE}$ will be unsatisfied. A two-dimensional example of a situation like that is shown in Figure 1. In Figure 1, the x -axis and the y -axis represent the first and second dimension of the state, respectively; the covariance ellipse of P^{LE} is the largest ellipse contained within the intersection region $\mathfrak{R}_{(\bar{x}_1, P_1)} \cap \mathfrak{R}_{(\bar{x}_2, P_2)}$, but $\bar{P}^{LE} \not\leq P^{LE}$, which implies that the fused result is inconsistent.

However, the covariance ellipsoid corresponding to the actual fused error covariance accounting for the dependence of local estimates will become smaller as the dependence becomes weaker [1], which means that the LE algorithm is likely to be consistent when the local estimates are weakly correlated. On the other hand, by comparing (16)–(19) to (6) and (8), we find that, unlike the optimal distributed Kalman fuser, which computes its state and error covariance using the same weighting matrices, the error covariance estimation of LE algorithm is independent of its state estimation. If the LE algorithm can be extended to the multisensor system, it is possible to raise its actual fused accuracy

and subsequently improve its consistency performance by taking full advantage of the information of each sensor.

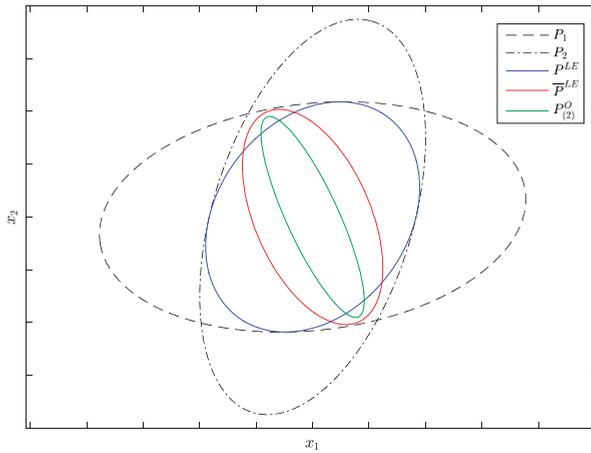


Figure 1. An accuracy relation example of the (largest ellipsoid) LE algorithm.

4. Multisensor Largest Ellipsoid Fusers

4.1. Multisensor Largest Ellipsoid Fuser with Sequential Structure

One way to extend the application of LE algorithm from a two-sensor case to a multisensor case is applying the sequential processing method as the SCI algorithm proposed in [18]; here, we label such multisensor LE fuser as a Sequential Largest Ellipsoid (SLE) fuser. By doing so, the multisensor LE fusion for multisensor system (1) consists of $L - 1$ sequential LE fusions. The structure of SLE fuser is shown schematically in Figure 2. In Figure 2, ‘KF’ represents the local steady-state Kalman filter of each subsystem, ‘LE’ represents a LE algorithm operation, and the green dashed lines or curves indicate the routes via which the local estimates are fused into the final fused result.

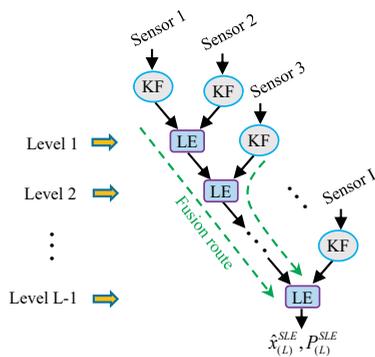


Figure 2. The structure of the SLE fuser.

The SLE fuser has a recursive sequential form as

$$\begin{aligned} \hat{x}_{i+1}^{SLE} &= A_{1/i}^{LE} \hat{x}_i^{SLE} + A_{2/i}^{LE} \hat{x}_{i+1} \\ P_{i+1}^{SLE} &= B_{1/i}^{LE} P_i^{SLE} + B_{2/i}^{LE} P_{i+1} \end{aligned}, \quad i = 1, 2, \dots, L - 1. \tag{20}$$

with initial values $P_1^{SLE} = P_1$, $x_1^{SLE} = \hat{x}_1$; $A_{1/i}^{LE}$ and $A_{2/i}^{LE}$ are the weighting matrices for state estimation corresponding to the LE fusion in the i th fusion level; $B_{1/i}^{LE}$ and $B_{2/i}^{LE}$ are the weighting matrices for error covariance estimation corresponding to the LE fusion in the i th fusion level. In addition, the fused estimate of SLE fuser is defined as

$$\begin{aligned} \hat{x}_{(L)}^{SLE} &= \hat{x}_L^{SLE} \\ P_{(L)}^{SLE} &= P_L^{SLE} \end{aligned} \quad (21)$$

Combining (20) with (21) to expand the recursive sequential form of the SLE fuser yields linear expressions as follows:

$$\begin{aligned} \hat{x}_{(L)}^{SLE} &= \sum_{i=1}^L A_i^{SLE} \hat{x}_i \\ P_{(L)}^{SLE} &= \sum_{i=1}^L B_i^{SLE} P_i \end{aligned} \quad (22)$$

where the weighting matrices are computed by

$$\begin{aligned} A_i^{SLE} &= A_{2/i-1}^{LE} \prod_{j=i}^{L-1} A_{1/j}^{LE} \\ B_i^{SLE} &= B_{2/i-1}^{LE} \prod_{j=i}^{L-1} B_{1/j}^{LE} \end{aligned} \quad (23)$$

with $A_{2/0}^{LE} = B_{2/0}^{LE} = I_n$. From (23), we know that both A_i^{SLE} and B_i^{SLE} are the multiplication results of all the LE fusion weighting matrices that the i th local estimate encounters in its fusion route.

4.2. Multisensor Largest Ellipsoid Fusers with Parallel Structures

As we can see from Figure 2, the sensors must be fused sequentially in the SLE fuser, which makes the SLE fuser inefficient in multiprocessor operations; in addition, the studies in the following part of this paper will show that the performance of SLE fuser is relatively poor. To handle such disadvantages, a largest ellipsoid fusion Kalman filtering with parallel structure for the data fusion of multisensor system, called a Parallel Largest Ellipsoid (PLE) fuser, is also proposed in this work. It realizes the data fusion of multisensor system (1) with a multilevel fusion and each fusion level consists of one or a series of parallel LE fusions. The structure of PLE fuser is shown schematically in Figure 3.

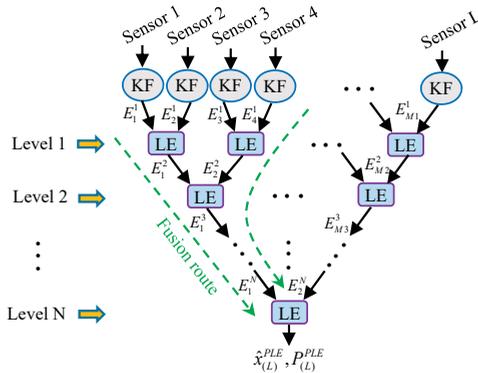


Figure 3. The structure of the PLE fuser.

The PLE fuser contains N fusion levels with N satisfying the inequation $2^{N-1} < L \leq 2^N$. In addition, the i th fusion level includes $\lfloor L/2^i \rfloor$ LE fusions and generates $\lceil L/2^i \rceil$ new fused estimates, where notations $\lfloor * \rfloor$ and $\lceil * \rceil$ denote rounding down and rounding up, respectively. The PLE fuser is realized by the following steps:

- Step 1: In the first fusion level, all of the local estimates received from local steady-state Kalman filters are fused in pairs using the LE algorithm. When the number of local estimates is even, we can get $\frac{L}{2}$ new fused estimates; and we can obtain $\frac{L-1}{2} + 1$ new fused estimates including an unsettled local estimate when the number of local estimates is odd. Then, the new fused estimates are passed to the next fusion level.
- Step 2: As Step 1, all the estimates received from the upper fusion level are fused in pairs using the LE algorithm, and the obtained new fused estimates are passed to the next fusion level.
- ⋮
- Step N: There are only two estimates received from the upper fusion level in the fusion level N and the fusion result of them through the LE algorithm is defined as the PLE fuser estimate $N(\hat{x}_{(L)}^{PLE}, P_{(L)}^{PLE})$.

As shown in Figure 3, we denote the received estimates in the i th fusion level of PLE fuser, respectively, as $E_1^i, E_2^i, \dots, E_{M_i}^i$ from the left side to the right side, where M_i represents the number of the received estimates in the i th fusion level. Notice that the received estimates in the fusion levels of PLE fuser can be paired by different methods that will lead to different types of PLE fusers with heterogeneous parallel fusion structures. In this paper, we give three estimate pairing methods as follows.

- Method 1:** In the i th fusion level, the received estimates are paired from E_1^i to $E_{M_i}^i$. If there is an unsettled received estimate in the i th fusion level, it must be $E_{M_i}^i$.
- Method 2:** The fusion levels of this type of PLE fuser alternately pair their received estimates from E_1^i to $E_{M_i}^i$ or from $E_{M_i}^i$ to E_1^i . For instance, the local estimates are paired from E_1^1 to $E_{M_1}^1$ in the fusion level 1, the received estimates in the fusion level 2 are paired from $E_{M_2}^2$ to E_1^2 , and the received estimates in the fusion level 3 are paired from E_1^3 to $E_{M_3}^3$, etc. If there is an unsettled received estimate in the i th fusion level, it must be E_1^i or $E_{M_i}^i$.
- Method 3:** In the i th fusion level, the received estimates $E_{M_i}^i$ and E_1^i are grouped into a pair with their fused estimate treated as E_1^{i+1} in the next fusion level, and the remaining received estimates are paired from E_2^i to $E_{M_i-1}^i$. If there is an unsettled received estimate in the i th fusion level, it must be $E_{M_i-1}^i$.

In the following part of this paper, the notations PLE1, PLE2 and PLE3 denote the PLE fusers, respectively, with estimate pairing Method 1, Method 2 and Method 3. For the multisensor system (1) consisting of five sensors, the fusion schemes of PLE1, PLE2 and PLE3 fusers are shown schematically in Figure 4. In Figure 4, the empty circle at a certain fusion level represents an estimate received from its upper fusion level, and it is the unsettled received estimate, which is directly passed to its next fusion level without fusing with other received estimates.

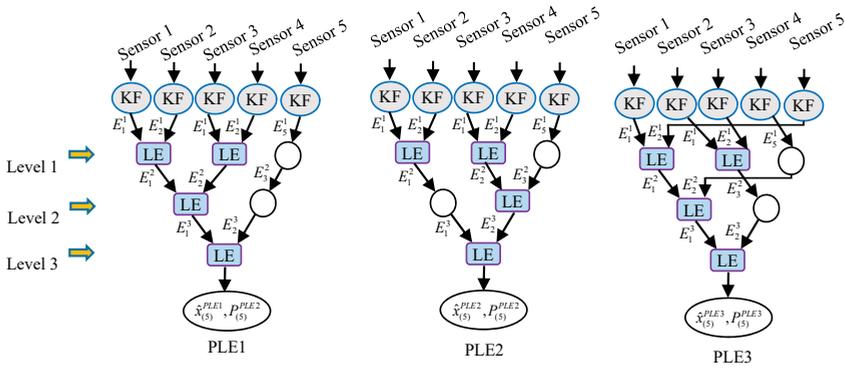


Figure 4. The fusion schemes of PLE1, PLE2 and PLE3 fusers with five sensors.

Similarly to the SLE fuser, the PLE fuser can also be formulated in linear expressions as follows:

$$\begin{aligned} \hat{x}_{(L)}^{PLE} &= \sum_{i=1}^L A_i^{PLE} \hat{x}_i \\ P_{(L)}^{PLE} &= \sum_{i=1}^L B_i^{PLE} P_i \end{aligned} \quad (24)$$

with weighting matrices computed by

$$\begin{aligned} A_i^{PLE} &= A_{i(N)}^{LE} \dots A_{i(2)}^{LE} A_{i(1)}^{LE} \\ B_i^{PLE} &= B_{i(N)}^{LE} \dots B_{i(2)}^{LE} B_{i(1)}^{LE} \end{aligned} \quad (25)$$

where $A_{i(j)}^{LE}, B_{i(j)}^{LE}$ indicate the LE fusion weighting matrices corresponding to the LE fusion that the i th local estimate encounters in the j th fusion level in its fusion route; both $A_{i(j)}^{LE}$ and $B_{i(j)}^{LE}$ will be equal to I_n if the estimate in the j th fusion level is unsettled in the fusion route.

4.3. Properties of Multisensor Largest Ellipsoid Fusers

Comparing (18) to (7), we can find that the state estimation weighting matrices of the LE algorithm will deviate from the optimal weighting matrices, which are the weighting matrices of the optimal distributed Kalman fuser having the same sensors as the LE algorithm, on account of the inaccurate error covariances of local estimates and the presence of unknown cross-covariances. Taking this point into account, the state estimation weighting matrices of SLE fuser and PLE fuser, respectively expressed in (23) and (25) can be rewritten into

$$\begin{aligned} A_i^{SLE} &= \left(A_{2/i-1}^O + \Delta A_{2/i-1} \right) \prod_{j=i}^{L-1} \left(A_{1/j}^O + \Delta A_{1/j} \right) \\ A_i^{PLE} &= \left(A_{i(N)}^O + \Delta A_{i(N)} \right) \dots \left(A_{i(2)}^O + \Delta A_{i(2)} \right) \left(A_{i(1)}^O + \Delta A_{i(1)} \right) \end{aligned} \quad (26)$$

with $A_{2/0}^O + \Delta A_{2/0} = I_n$. $A_{1/i'}^O, A_{2/j'}$, $i = 1, 2, \dots, L - 1$ being the optimal weighting matrices corresponding to the LE fusion in the i th fusion level in the SLE fuser, and $A_{i(j)}^O$ being the optimal weighting matrix corresponding to the LE fusion that the i th local estimate encounters in the j th fusion level in its fusion route in the PLE fuser; $\Delta A_{1/i}, \Delta A_{2/i'}$, $i = 1, 2, \dots, L - 1$ and $\Delta A_{i(j)}$ are the weighting matrix errors that the state estimation weighting matrices of these LE fusions deviate from their corresponding optimal weighting matrices. Hence, given a multisensor system, the number of the LE fusions that each local estimate encounters in its fusion route and how these numbers differ

from each other will affect the weight assignments for the local estimates in the fuser, which implies that the fuser structure has a significant influence on the characteristic and performance of the fuser. From Figures 2 and 3, we see that the number of the LE fusions that each local estimate encounters in its fusion route can be different in the SLE fuser and the PLE fusers based on different estimate pairing methods. In order to give a further analysis of the features of SLE fuser and PLE fuser for multisensor system (1), here we define two fusion performance assessment parameters as Fusion Distance and Fusion Index.

Definition 1. The Fusion Distance $D_{(i)}^{(j)}$ indicates the number of the LE fusions that the i th local estimate encounters in its fusion route in fuser j (SLE, PLE1, PLE2 or PLE3).

Remark 1. When $L \geq 2$, we have $\max(D_{(i)}^{(SLE)}) = L - 1$ and $\min(D_{(i)}^{(SLE)}) = 1$. For any certain $N \geq 1$, we have $\max(D_{(i)}^{(PLE1)}) = \max(D_{(i)}^{(PLE2)}) = \max(D_{(i)}^{(PLE3)}) = N$, $\min(D_{(i)}^{(PLE1)}) = 1$, $\min(D_{(i)}^{(PLE2)}) = N - \lceil \frac{N-1}{2} \rceil$, and $\min(D_{(i)}^{(PLE3)}) = \max(N - 1, 1)$.

Definition 2. The Fusion Index $F^{(j)}$ shows to what extent the Fusion Distances of all the local estimates differ from each other in fuser j (SLE, PLE1, PLE2 or PLE3). It is defined as

$$F^{(j)} = \max \left(D_{(i)}^{(j)} \right)_{i=1,2,\dots,L} - \min \left(D_{(i)}^{(j)} \right)_{i=1,2,\dots,L}. \tag{27}$$

Remark 2. When $L \geq 2$, we have $F^{(SLE)} = L - 2$. For any certain $N \geq 1$, we have $\max(F^{(PLE1)}) = N - 1$, $\max(F^{(PLE2)}) = \lceil \frac{N-1}{2} \rceil$, $\max(F^{(PLE3)}) = \min(N - 1, 1)$.

Because the fused error covariance ellipsoid of LE algorithm is contained within the intersection region of the covariance ellipsoids of local estimates, we can easily come to the conclusion that $P_{(L)}^{SLE} \leq P_i, P_{(L)}^{PLE} \leq P_i, i = 1, 2, \dots, L$. However, as we can see from (23) and (25), the error covariance estimation weighting matrices of SLE fuser and different PLE fusers are varied with the structure, hence the fused error covariances of these fusers are generally different from each other. For multisensor system (1), when adding a new sensor to the system, we apparently have $P_{(L+1)}^{SLE} \leq P_{(L)}^{SLE}$. However, the situation of the PLE fuser is more complicated, but it is obvious that if the existing fusion structure of a PLE fuser is not affected by the new sensor, then the one PLE fuser will embrace $P_{(L+1)}^{PLE} \leq P_{(L)}^{PLE}$, such as the PLE1 fuser. The fused accuracies of SLE fuser and PLE fuser with such property will become higher and higher as the number of fused sensors increases. In the sense that both $\hat{x}_{(L)}^O, \hat{x}_{(L)}^{SLE}$ and $\hat{x}_{(L)}^{PLE}$ are the linear unbiased estimates of state x and $\hat{x}_{(L)}^O$ is the LUMV estimate, we have $P_{(L)}^O \leq \bar{P}_{(L)}^{SLE}$ and $P_{(L)}^O \leq \bar{P}_{(L)}^{PLE}$, where $\bar{P}_{(L)}^{SLE} = E \left[(\hat{x}_{(L)}^{SLE} - x)(\hat{x}_{(L)}^{SLE} - x)^T \right]$ and $\bar{P}_{(L)}^{PLE} = E \left[(\hat{x}_{(L)}^{PLE} - x)(\hat{x}_{(L)}^{PLE} - x)^T \right]$ are the actual fused error covariances of the SLE fuser and PLE fuser, respectively. In addition, we can achieve the upper bounds for the actual fused error covariances of the SLE fuser and PLE fuser irrespective of the cross-covariances. For multisensor system (1), according to [24], for any factors $\infty \geq \rho_i \geq 1, i = 1, 2, \dots, L$, when

$$\sum_{i=1}^L \frac{1}{\rho_i} = 1, \tag{28}$$

we have

$$\Sigma \leq \begin{bmatrix} \rho_1 P_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \rho_L P_L \end{bmatrix}, \tag{29}$$

For the SLE fuser and PLE fuser, we obtain

$$\begin{aligned} \bar{P}_{(L)}^k &= [A_1^k, \dots, A_L^k] \Sigma [A_1^k, \dots, A_L^k]^T \\ &\leq [A_1^k, \dots, A_L^k] \begin{bmatrix} \rho_1 P_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \rho_L P_L \end{bmatrix} [A_1^k, \dots, A_L^k]^T, \\ &= \sum_{i=1}^L \rho_i A_i^k P_i (A_i^k)^T, \quad k \in (SLE, PLE) \end{aligned} \tag{30}$$

By taking the minimization of error covariance trace as the optimization target, we get the optimization model as

$$\begin{aligned} \min & \left\{ \sum_{i=1}^L \rho_i \text{tr} \left[A_i^k P_i (A_i^k)^T \right] \right\} \\ \text{s.t.} & \sum_{i=1}^L \frac{1}{\rho_i} = 1 \end{aligned} \tag{31}$$

Applying the Lagrange multiplier method, we introduce the Lagrange function defined by

$$f(\rho_1, \dots, \rho_L, \lambda) = \sum_{i=1}^L \rho_i \text{tr} \left[A_i^k P_i (A_i^k)^T \right] + \lambda \left(\sum_{i=1}^L \frac{1}{\rho_i} - 1 \right), \tag{32}$$

where λ is the Lagrange multiplier. Then, we can achieve

$$\rho_i = \frac{\sum_{j=1}^L \sqrt{\text{tr} \left[A_j^k P_j (A_j^k)^T \right]}}{\sqrt{\text{tr} \left[A_i^k P_i (A_i^k)^T \right]}}. \tag{33}$$

Replacing (33) into (30), we obtain

$$\bar{P}_{(L)}^k \leq \sum_{i=1}^L \frac{\sum_{j=1}^L \sqrt{\text{tr} \left[A_j^k P_j (A_j^k)^T \right]}}{\sqrt{\text{tr} \left[A_i^k P_i (A_i^k)^T \right]}} A_i^k P_i (A_i^k)^T, \quad k \in (SLE, PLE). \tag{34}$$

5. Simulations and Analysis

5.1. Simulations

Consider a dynamic example of the multisensor system (1) with five sensors as

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0.5T^2 \\ T \end{bmatrix} w(t), \\ y_i(t) &= H_i x(t) + v_i(t), \quad i = 1, 2, \dots, 5 \end{aligned} \tag{35}$$

where $T = 0.5$ is the sample period, $t = 1, 2, \dots, 300$ is the discrete time (step), $x_0 = [10, 2]^T$ is the initial state, $Q = 2$, and

$$\begin{aligned} H_1 &= I_2, R_1 = \text{diag}(7.0, 0.22) \\ H_2 &= I_2, R_2 = \text{diag}(2.85, 0.3) \\ H_3 &= I_2, R_3 = \text{diag}(1.3, 1.5) \\ H_4 &= I_2, R_4 = \text{diag}(0.55, 3.1) \\ H_5 &= [1, 0], R_5 = 0.6 \end{aligned} \tag{36}$$

The local steady-state Kalman estimates, local filtering error covariances and cross-covariances can be obtained according to (2)–(5), and then the fused estimates of the optimal distributed Kalman fuser weighted by matrices, the CI algorithm, the SLE fuser and PLE fusers can be computed. The fusion schemes of PLE1, PLE2 and PLE3 fusers are shown in Figure 4. The traces of the theoretical error covariance matrices of local and fused estimates are listed in Table 1, and their corresponding covariance ellipses are illustrated in Figure 5. In Figure 5, the center of each covariance ellipse is $x(t)$, and the x -axis and the y -axis indicate how far the covariance ellipses extend in the directions of the first and second dimension of the state from the center, respectively.

Table 1. The traces of theoretical error covariance matrices.

$\text{tr}(P_1)$	$\text{tr}(P_2)$	$\text{tr}(P_3)$	$\text{tr}(P_4)$	$\text{tr}(P_5)$	$\text{tr}(P_{(5)}^O)$	$\text{tr}(\bar{P}_{(5)}^{CI})$	$\text{tr}(P_{(5)}^{CI})$
0.7433	0.6155	1.0032	0.8962	1.1932	0.1812	0.3233	0.5863
$\text{tr}(\bar{P}_{(5)}^{SLE})$	$\text{tr}(P_{(5)}^{SLE})$	$\text{tr}(\bar{P}_{(5)}^{PLE1})$	$\text{tr}(P_{(5)}^{PLE1})$	$\text{tr}(\bar{P}_{(5)}^{PLE2})$	$\text{tr}(P_{(5)}^{PLE2})$	$\text{tr}(\bar{P}_{(5)}^{PLE3})$	$\text{tr}(P_{(5)}^{PLE3})$
0.2976	0.3861	0.2550	0.3860	0.2244	0.3861	0.2217	0.3876

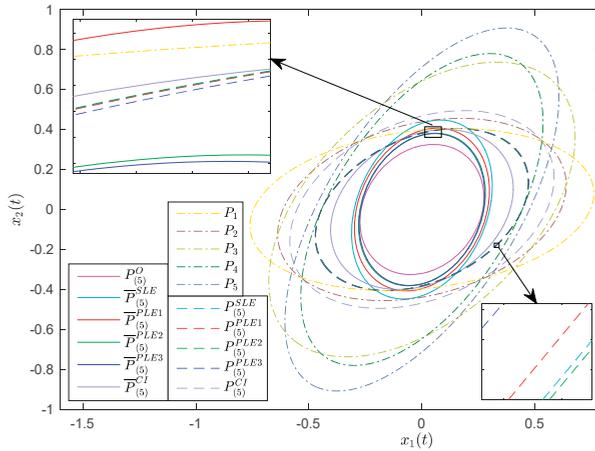


Figure 5. The error covariance ellipses of local and fused estimates.

From Table 1, we see that both $\text{tr}(\bar{P}_{(5)}^{SLE})$ and $\text{tr}(\bar{P}_{(5)}^{PLEi})$, $i = 1, 2, 3$ are greater than but close to $\text{tr}(P_{(5)}^O)$, which imply that the actual fused accuracies of SLE fuser and PLE fusers are close to that of the optimal fuser. The $\text{tr}(P_{(5)}^{SLE})$ and $\text{tr}(P_{(5)}^{PLEi})$, $i = 1, 2, 3$ are almost the same and obviously less than $\text{tr}(P_{(5)}^{CI})$ and $\text{tr}(P_i)$, $i = 1, 2, \dots, 5$, which mean that the fused accuracies of SLE fuser and PLE fusers are approximately equal and higher than those of the CI algorithm and each local estimate. Note that, although the $\text{tr}(\bar{P}_{(5)}^{SLE})$ and $\text{tr}(\bar{P}_{(5)}^{PLEi})$, $i = 1, 2, 3$ are, respectively, less than $\text{tr}(P_{(5)}^{SLE})$ and $\text{tr}(P_{(5)}^{PLEi})$, $i = 1, 2, 3$, it can not be concluded that the SLE fuser and PLE fusers are consistent due to fact that the LE algorithm can not guarantee its consistency. Correspondingly, in Figure 5, the covariance ellipse of $P_{(5)}^O$ is enclosed in the covariance ellipses of $\bar{P}_{(5)}^{SLE}$ and $\bar{P}_{(5)}^{PLEi}$, $i = 1, 2, 3$, which indicates $P_{(5)}^O \leq \bar{P}_{(5)}^{SLE}$ and $P_{(5)}^O \leq \bar{P}_{(5)}^{PLEi}$, $i = 1, 2, 3$. The covariance ellipses of $P_{(5)}^{SLE}$ and $P_{(5)}^{PLEi}$, $i = 1, 2, 3$ are enclosed in the covariance ellipses of $P_{(5)}^{CI}$ and P_i , $i = 1, 2, \dots, 5$, which indicate $P_{(5)}^{SLE} \leq P_i$, $i = 1, 2, \dots, 5$, $P_{(5)}^{SLE} \leq P_{(5)}^{CI}$, $P_{(5)}^{PLEj} \leq P_i$, $i = 1, 2, \dots, 5$; $j = 1, 2, 3$ and $P_{(5)}^{PLEj} \leq P_{(5)}^{CI}$, $j = 1, 2, 3$.

The covariance ellipses of $P_{(5)}^{SLE}$ and $P_{(5)}^{PLEi}$, $i = 1, 2, 3$ are almost overlapping, which implies that the SLE fuser and PLE fusers each obtain a similar fused error covariance. However, the covariance ellipse of $\bar{P}_{(5)}^{SLE}$ is not enclosed in the covariance ellipses of $P_{(5)}^{SLE}$ and P_1 , which means that $\bar{P}_{(5)}^{SLE} \not\leq P_{(5)}^{SLE}$ and $\bar{P}_{(5)}^{SLE} \not\leq P_1$; and the covariance ellipse of $\bar{P}_{(5)}^{PLE1}$ is not enclosed in the covariance ellipses of $P_{(5)}^{PLE1}$ and P_1 , which means $\bar{P}_{(5)}^{PLE1} \not\leq P_{(5)}^{PLE1}$ and $\bar{P}_{(5)}^{PLE1} \not\leq P_1$; thus, the SLE fuser and PLE1 fuser are inconsistent in this example. While the covariance ellipses of $\bar{P}_{(5)}^{PLE2}$ and $\bar{P}_{(5)}^{PLE3}$ are, respectively, enclosed in the covariance ellipses of $P_{(5)}^{PLE2}$ and $P_{(5)}^{PLE3}$, which mean $\bar{P}_{(5)}^{PLE2} \leq P_{(5)}^{PLE2}$ and $\bar{P}_{(5)}^{PLE3} \leq P_{(5)}^{PLE3}$, the PLE2 and PLE3 fusers are consistent here.

In order to verify the above theoretical results on the accuracy relation, the Monte Carlo method is applied to compute mean square error (MSE). The MSE value at time t for local or fused estimate \hat{x}^i with error covariance P^i is defined as

$$MSE_{N_{run}}^i(t) = \frac{1}{N_{run}} \sum_{j=1}^{N_{run}} [\hat{x}_j^i(t|t) - x_j(t)]^T [\hat{x}_j^i(t|t) - x_j(t)], \tag{37}$$

where N_{run} is the number of Monte Carlo runs, $\hat{x}_j^i(t|t)$ and $x_j(t)$ denote the j th realization of $\hat{x}^i(t|t)$ and $x(t)$, respectively. Because

$$\text{tr}(P^i) = \text{tr} \left\{ E \left[(\hat{x}^i - x) (\hat{x}^i - x)^T \right] \right\} = E \left[(\hat{x}^i - x)^T (\hat{x}^i - x) \right], \tag{38}$$

according to the ergodicity [25], we have

$$MSE_{N_{run}}^i(t) = \text{tr}(P^i), \quad t \rightarrow \infty, N_{run} \rightarrow \infty. \tag{39}$$

For the dynamic example (35)–(36), 1000 Monte Carlo runs are performed, and the statistical results of local estimates and the fused estimates of optimal fuser, SLE fuser and PLE fusers in the Monte Carlo simulation are shown in Figure 6.

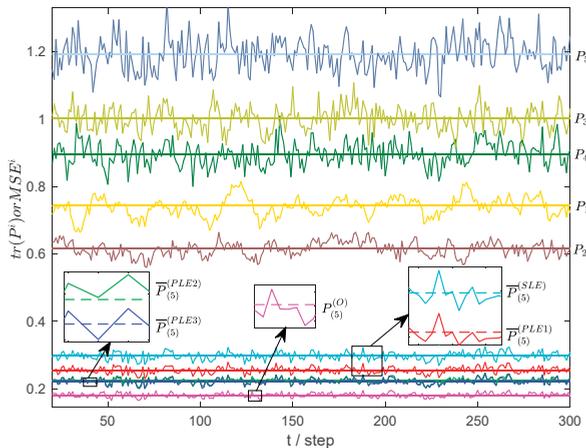


Figure 6. The $\text{tr}(P^i)$ and MSE^i of local and fused estimates.

In Figure 6, the straight lines and dashed lines denote $\text{tr}(P^i)$, and the solid curves denote $MSE_{N_{run}}^i$; P^i represents P_i , $i = 1, 2, \dots, 5$, $P_{(5)}^O$, $\bar{P}_{(5)}^{SLE}$ and $\bar{P}_{(5)}^{PLEi}$, $i = 1, 2, 3$, while $MSE_{N_{run}}^i$ denotes

their corresponding MSE values. From Figure 6, we know that the $MSE_{N_{run}}^i$ fluctuates around $\text{tr}(P^i)$, which is consistent with (39); and the statistical accuracy relations of local and fused estimates indicated by $MSE_{N_{run}}^i$ are coincident with the theoretical results shown in Table 1.

As shown in Figures 2 and 3, both the SLE fuser and PLE fuser schemes will vary as long as the sensor order varies. To explore how the accuracies of SLE fuser and the PLE fusers based on different estimate pairing methods are related to the sensor orders, all the sensor permutations are considered and simulated. The accuracies of SLE fuser and PLE fusers with respect to different sensor orders are given in Figure 7. In Figure 7, the x -axis represents all of the permutations of five used sensors, 120 in total; the y -axis represents the traces of $P_{(5)}^O, \bar{P}_{(5)}^{SLE}, \bar{P}_{(5)}^{PLEi}, i = 1, 2, 3$ and $P_{(5)}^{SLE}, P_{(5)}^{PLEi}, i = 1, 2, 3$.

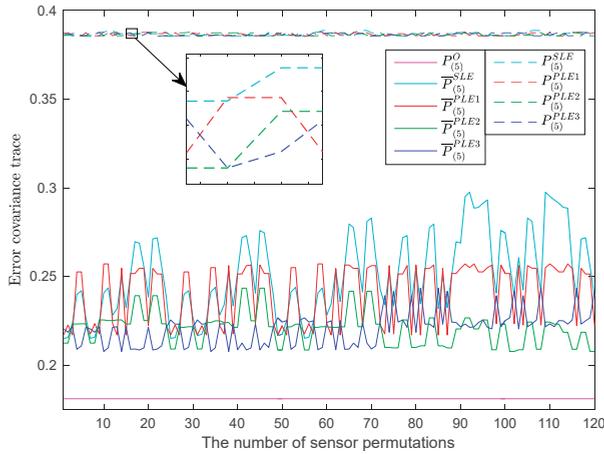


Figure 7. The fuser accuracies with respect to different orders of five sensors.

Figure 7 shows that, for all possible sensor orders, the fused accuracies of SLE fuser and PLE fusers are almost the same, while their actual fused accuracies are more different, which mean that the fused accuracies of SLE fuser and PLE fusers are less affected by the sensor orders, but their actual fused accuracies are greatly influenced. In order to strengthen the information shown in Figure 7, another four sensors are added to systems (35) and (36), and they are given as

$$\begin{aligned}
 H_6 &= I_2, R_6 = \text{diag}(2.1, 2.06) \\
 H_7 &= I_2, R_7 = \text{diag}(1.1, 7.56) \\
 H_8 &= I_2, R_8 = \text{diag}(16.6, 0.15) \\
 H_9 &= I_2, R_9 = \text{diag}(0.9, 23.0)
 \end{aligned} \tag{40}$$

In addition, the traces of the theoretical error covariance matrices of their local steady-state Kalman estimates are shown in Table 2.

Table 2. The traces of the theoretical error covariance matrices of added local estimates.

$\text{tr}(P_6)$	$\text{tr}(P_7)$	$\text{tr}(P_8)$	$\text{tr}(P_9)$
1.3512	1.3910	0.8807	1.3865

The accuracies of SLE fuser and PLE fusers with respect to different sensor orders for the expanded multisensor system with nine sensors are shown in Figure 8. In Figure 8, the x -axis represents all of the

permutations of nine used sensors, 362,880 in total; and the legends of this figure are the same as those in Figure 7.

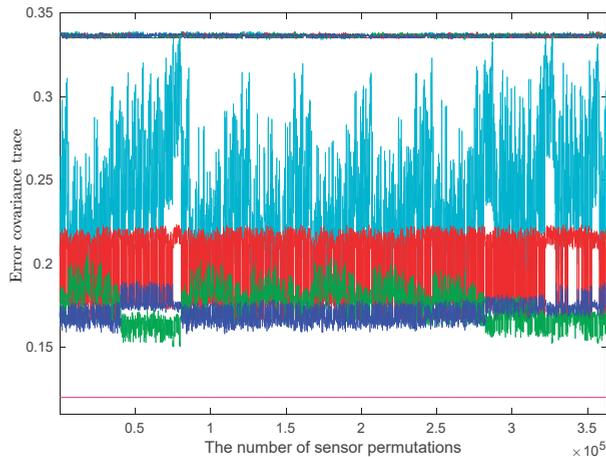


Figure 8. The fuser accuracies with respect to different orders of nine sensors.

From Figures 7 and 8, it can be seen that with a certain number of sensors, no matter how the sensors are ordered, the $\text{tr}(P^{SLE})$ and $\text{tr}(P^{PLEi})$, $i = 1, 2, 3$ are almost equal, which mean that the fused accuracies of SLE fuser and PLE fusers are almost equivalent and are insensitive to the sensor orders. When the multisensor system only has a few sensors, the actual fused accuracies of SLE fuser and PLE fusers perform approximately and have similar sensitivity to sensor orders. However, as the number of sensors increases, the differences among them are becoming increasingly significant. From Figure 8, we see that the $\text{tr}(\bar{P}^{SLE})$ fluctuates most drastically and is generally greater than $\text{tr}(\bar{P}^{PLEi})$, $i = 1, 2, 3$, which means that the actual fused accuracy of SLE fuser is poorer and more sensitive to the sensor orders than these of PLE fusers. Whether in actual fused accuracy or in sensitivity to the sensor orders, the PLE2 and PLE3 fusers perform better than the PLE1 fuser, and the PLE3 fuser generally has the best performance.

In practice, the number of used sensors may vary in different periods. For the subsystems individually with sensors $1 \sim i$, $i = 1, 2, \dots, 9$, the accuracies of the SLE fuser and PLE fusers with sensors fused in normal order are presented in Figure 9. In Figure 9, the number i on the x -axis not only represents the i th subsystem using the sensors $1 \sim i$, but also represents the i th sensor. From Figure 9, we know that the actual fused accuracy of SLE fuser is less robust to the accuracy of the newly added sensor than these of PLE fusers; and, compared to the PLE1 fuser, the actual fused accuracies of PLE2 and PLE3 fusers perform more robustly and they tend to become higher as the number of sensors increases. Furthermore, the accuracies of PLE fusers are significantly better than that of the CI algorithm.

In the simulation cases above, we have specified the characteristics of the sensors. Without such specifications, we give a further study on the performances of the SLE fuser and PLE fusers in the multisensor systems with arbitrary overall error covariances. Because the overall error covariance Σ of the multisensor system (1) is a real symmetric positive definite matrix, it has diagonal decomposition as $\Sigma = \Theta \Xi \Theta^T$, where Θ is an orthogonal matrix, Ξ is a diagonal matrix. Θ and Ξ can be randomly created using Matlab (R2015b, MathWorks, Natick, MA, US) functions, such as 'orth', 'diag' and 'randn', etc. For the multisensor system consisting of nine two-dimensional sensors, 30 random overall error covariance matrices are simulated with each element of their Ξ selected from the positive samples of a random variable, which is of standard normal distribution, and the fused results of SLE fuser

and PLE fusers are shown in Figure 10. In Figure 10, the x -axis represents the 30 simulated random overall error covariances, and the legends of this figure are the same as these in Figure 9. As shown in Figure 10, for most of the simulated overall error covariance matrices, the multisensor LE fusers sorted in descending order of actual fused accuracy are PLE3, PLE2, PLE1 and the SLE fuser, and they all have higher accuracies than the CI algorithm.

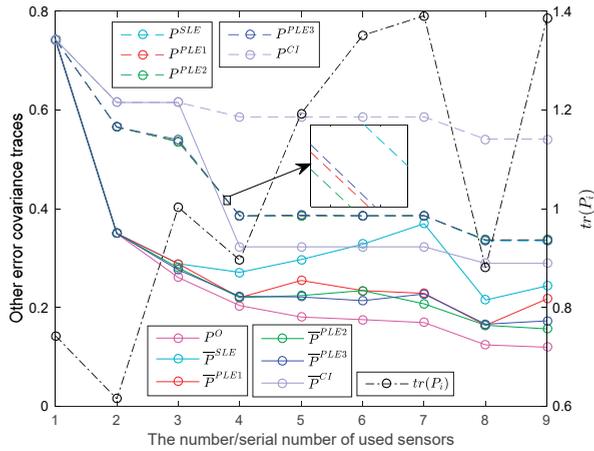


Figure 9. The fuser accuracies with respect to different numbers of used sensors.

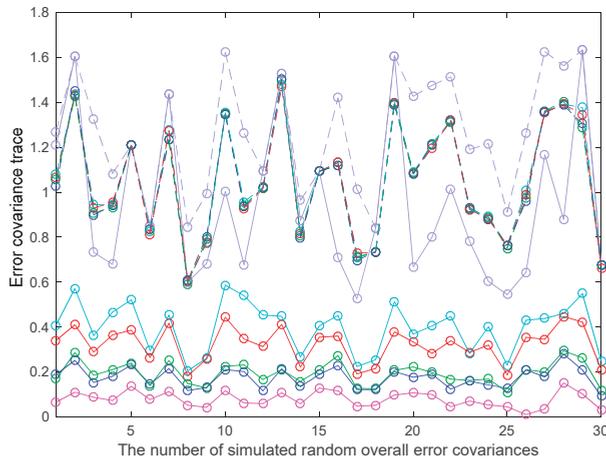


Figure 10. The fuser accuracies with respect to different random overall error covariance matrices.

For any two local estimates \hat{x}_i and \hat{x}_j in the multisensor system, the correlation property of them can be measured by the following correlation model [11]:

$$P_{ij} = \gamma J_i J_j^T, \tag{41}$$

where $\gamma \in [0, 1)$ is the correlation coefficient between \hat{x}_i and \hat{x}_j ; J_k , $k = i, j$ is the Cholesky decomposition of P_k satisfying $J_k J_k^T = P_k$. As described above, the inconsistency of the LE algorithm is resulted from the unknown cross-covariances among local estimates. In order to investigate how the

consistencies of SLE fuser and PLE fusers are related to the unknown cross-covariances, the multisensor systems consisting of nine two-dimensional sensors under different correlation coefficients are studied. Meanwhile, in order to cover the main range of the correlation coefficient, in this case, each studied correlation coefficient is given by $\gamma_i = 0.01n_i$, where the integer $n_i \in [0, 99]$ is the serial number of the correlation coefficients. For each performed correlation coefficient γ_i , 200 random overall error covariance matrices of the multisensor system are firstly created, and then in each overall error covariance matrix, the cross-covariance between any two local estimates is replaced by the new cross-covariance obtained using (41). The consistency ratios of the number of consistent fused results to the total number of fused results for SLE fuser and PLE fusers with respect to different correlation coefficients are shown in Figure 11.

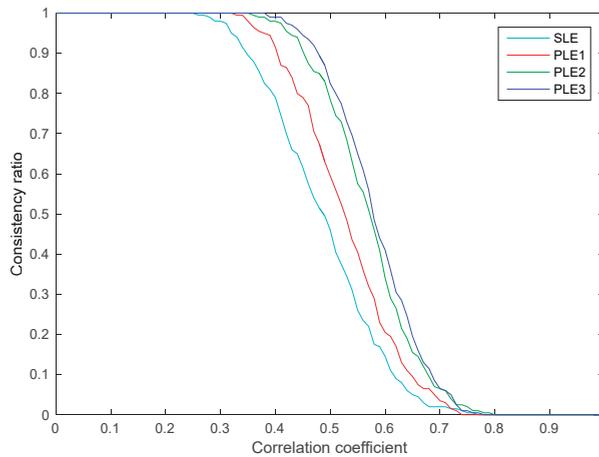


Figure 11. The consistency ratios with respect to different correlation coefficients.

From Figure 11, we know that the presented multisensor LE fusers are inconsistent when the local estimates are strongly correlated but are consistent when the local estimates are weakly correlated. Under a moderate correlation level, the multisensor LE fusers sorted in a descending order of the consistency ratio are PLE3, PLE2, PLE1 and SLE fusers. For certain moderate γ_i , compared to the SLE fuser, the PLE fusers significantly improve the consistency performance.

In summary, according to above simulation results, whether in the actual fused accuracy as well as its sensitivity to the sensor orders and its robustness to the accuracy of a newly added sensor, or in consistency and in computation efficiency, the PLE fusers have better performances than SLE fuser, and PLE3 fuser outperforms PLE2 fuser, which performs better than the PLE1 fuser.

5.2. Analysis

In order to obtain a fused estimate with high actual fused accuracy in a multisensor LE fuser, the weights of the state estimation weighting matrices should be assigned in accordance with the accuracies of sensors. As shown in (23) and (25), both A_i^{SLE} in SLE fuser and A_i^{PLE} in PLE fuser are the multiplication results of all of the LE fusion state estimation weighting matrices that the i th local estimate encounters in its fusion route. From (18), we have $A_1^{LE} < I_n$ and $A_2^{LE} < I_n$, the multiplication effect of multiple A_1^{LE} and (or) A_2^{LE} implies that the more LE fusions one local estimate encounters in its fusion route in the SLE fuser or PLE fuser, the less weight its weighting matrix A_i^{SLE} or A_i^{PLE} will tend to be. Since the number of the LE fusions that each local estimate encounters in its fusion route is affected by the fuser structure, the weights of the weighting matrices not only depend on the accuracies of sensors, but also are influenced by fuser structure. In addition, according to (26), the number of

LE fusions that each local estimate encounters in its fusion route also affects the weighting matrix deviations from the optimal weighting matrices.

Therefore, longer Fusion Distance means less weight and greater deviation; the Fusion Distance differences in the fuser affect the balances of weight assignment and deviation among local estimates and thus have a significant influence on the fuser's actual fused accuracy performance. Owing to the sequential structure, the Fusion Distances of each local estimate in the SLE fuser are remarkably different from each other, which leads to the fuser structure severely degrading the dependency of the actual fused accuracy on the accuracies of sensors. Comparatively, the differences of the local estimate Fusion Distances in PLE fuser are significantly reduced, which means that the actual fused accuracy of PLE fuser is more dependent on the accuracies of sensors and hence is better than that of the SLE fuser. The Fusion Index of the multisensor LE fuser not only represents the fuser's actual fused accuracy, but also indicates the sensitivity of the actual fused accuracy to the sensor orders and the robustness of the actual fused accuracy to the accuracy of the newly added sensor. A smaller Fusion Index means that the fuser has better performances in these aspects. In the simulation examples, when the multisensor system has five sensors, we have $F^{(SLE)} = 3$, $F^{(PLE1)} = 2$, $F^{(PLE2)} = F^{(PLE3)} = 1$, the PLE3, PLE2 fusers have the same level of performance in actual fused accuracy, sensitivity and robustness, and perform better than the SLE fuser and PLE1 fuser. When the multisensor system has nine sensors, we have $F^{(SLE)} = 7$, $F^{(PLE1)} = 3$, $F^{(PLE2)} = 2$, $F^{(PLE3)} = 1$, and the performance differences among SLE, PLE1, PLE2, PLE3 fusers become more significant. As the fused error covariances of the SLE fuser and PLE fuser are almost the same, then, for the same multisensor system, the fuser that has higher actual fused accuracy will have less possibility to become inconsistent. Accordingly, compared to the SLE fuser, the PLE fuser has better consistency performance, and the PLE2 fuser has worse performance than the PLE3 fuser but has better performance than the PLE1 fuser.

Apparently, compared to the SLE fuser, another one of the advantages of the PLE fuser is that the PLE fuser can make full use of the multiprocessor to fuse multiple sensors in parallel and more efficiently. Because each LE fusion can reduce the estimates to be fused by one in the SLE fuser and PLE fuser, it can be concluded that both the SLE fuser and the PLE fuser for the multisensor system (1) contain $L - 1$ LE fusions, and thus they have the same computation complexity. However, if multiprocessor parallel operation is used in data processing, their time complexity is proportional to their number of fusion levels. Therefore, the PLE fuser mostly outperforms the SLE fuser in computation efficiency. When the sensors are in a clustering distribution, parallel structure can also cut down the communication requirements of sensor networks.

6. Conclusions

For the multisensor system with unknown cross-covariances, this paper proposes a largest ellipsoid fusion Kalman filtering with parallel structure and gives three different estimate pairing methods to construct the parallel fusion structure. Two fusion performance assessment parameters of Fusion Distance and Fusion Index are defined, and the attributes of the SLE fuser and PLE fusers in Fusion Distance, Fusion Index and accuracy relation are given. Verified with examples, if a local estimate has a longer Fusion Distance, its weighting matrix will tend to be lighter in weight and of greater deviation. The Fusion Index reflects the influence of fusion structure on fusion performance and indicates a fuser's actual fused accuracy performance. A smaller Fusion Index implies that the actual fused accuracy of the fuser is generally higher, and is less sensitive to the sensor orders and more robust to the accuracy of newly added sensors. The presented multisensor LE fusers can achieve consistent fused results when the local estimates are weakly correlated but become inconsistent when the local estimates are strongly correlated; under such strong correlations, the upper bounds of actual fused error covariances of the presented multisensor LE fusers can be obtained by the provided formula to limit the uncertainties of the fused results. Compared to SLE fusers, the proposed PLE fusers not only can operate in parallel and more efficiently, but also get better performances in regards to accuracy and consistency.

Acknowledgments: This work is supported by the National Natural Science Foundation of China (Grant No. 11372177).

Author Contributions: Baoyu Liu conceived the main idea, performed the main part of the experiments and wrote the paper; Xingqun Zhan and Zheng H. Zhu critically reviewed the paper, contributed to the revision of the paper and provided important suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bar-Shalom, Y.; Campo, L. The effect of the common process noise on the two-sensor fused-track covariance. *IEEE Trans. Aerosp. Electron. Syst.* **1986**, *22*, 803–805. [CrossRef]
2. Kim, K.H. Development of Track to Track Fusion Algorithms. In Proceedings of the American Control Conference, Baltimore, MD, USA, 29 June–1 July 1994; pp. 1037–1041.
3. Sun, S.L.; Deng, Z.L. Multi-sensor optimal information fusion kalman filter. *Automatica* **2004**, *40*, 1017–1023. [CrossRef]
4. Lewis, F.L. *Optimal Estimation: With an Introduction to Stochastic Control Theory*; John Wiley & Sons: New York, NY, USA, 1986.
5. Sijs, J.; Lazar, M. State fusion with unknown correlation: Ellipsoidal intersection. *Automatica* **2012**, *48*, 1874–1878. [CrossRef]
6. Benaskeur, A.R. Consistent Fusion of Correlated Data Sources. In Proceedings of the Annual Conference of the Industrial Electronics Society, Sevilla, Spain, 5–8 November 2002; pp. 2652–2656.
7. Julier, S.J.; Uhlmann, J.K. A Non-Divergent Estimation Algorithm in the Presence of Unknown Correlations. In Proceedings of the American Control Conference, Albuquerque, NM, USA, 4–6 June 1997; pp. 2369–2373.
8. Zhou, Y.; Li, J. Data Fusion of Unknown Correlations Using Internal Ellipsoidal Approximation. In Proceedings of the International Federation of Automatic Control, Seoul, Korea, 6–11 July 2008; pp. 2856–2860.
9. Niehsen, W. Information Fusion Based on Fast Covariance Intersection Filtering. In Proceedings of the International Conference on Information Fusion, Annapolis, MD, USA, 8–11 July 2002; pp. 901–904.
10. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *14*, 28–44. [CrossRef]
11. Zhu, H.; Zhai, Q.; Yu, M.; Han, C. Estimation fusion algorithms in the presence of partially known cross-correlation of local estimation errors. *Inf. Fusion* **2014**, *18*, 187–196. [CrossRef]
12. Julier, S.J.; Uhlmann, J.K. General Decentralized Data Fusion with Covariance Intersection (CI). In *Handbook of Multisensor Data Fusion*; Llinas, J., David, L.H., Eds.; CRC Press: Boca Raton, FL, USA, 2001; pp. 12.1–12.25.
13. Wang, Y.; Li, X.R. Distributed estimation fusion with unavailable cross-correlation. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 259–278. [CrossRef]
14. Franken, D.; Hupper, A. Improved Fast Covariance Intersection for Distributed Data Fusion. In Proceedings of the International Conference on Information Fusion, Philadelphia, PA, USA, 25–28 July 2005; pp. 154–160.
15. Uhlmann, J.K. Covariance consistency methods for fault-tolerant distributed data fusion. *Inf. Fusion* **2003**, *4*, 201–215. [CrossRef]
16. Wang, Y.; Li, X.R. Distributed estimation fusion under unknown cross-correlation: An analytic center approach. In Proceedings of the International Conference on Information Fusion, Edinburgh, UK, 26–29 July 2010; pp. 1–8.
17. Farrell, W.J.; Ganesh, C. Generalized Chernoff Fusion Approximation for Practical Distributed Data Fusion. In Proceedings of the International Conference on Information Fusion, Seattle, WA, USA, 6–9 July 2009; pp. 555–562.
18. Deng, Z.; Zhang, P.; Qi, W.; Liu, J.; Gao, Y. Sequential covariance intersection fusion kalman filter. *Inf. Sci.* **2012**, *189*, 293–309. [CrossRef]
19. Wang, J.; Gao, Y.; Ran, C.; Huo, Y. State Estimation with Two-Level Fusion Structure. In Proceedings of the International Conference on Estimation, Detection and Information Fusion, Harbin, China, 10–11 January 2015; pp. 105–109.
20. Nargess, S.N.; Poshtan, J.; Wagner, A.; Nordheimer, E.; Badreddin, E. Cascaded kalman and particle filters for photogrammetry based gyroscope drift and robot attitude estimation. *ISA Trans.* **2014**, *53*, 524–532.

21. Bernstein, D.S. *Matrix Mathematics: Theory, Facts, and Formulas*, 2nd ed.; Princeton University Press: Princeton, NJ, USA, 2009.
22. Deng, Z.; Zhang, P.; Qi, W.; Yuan, G.; Liu, J. The accuracy comparison of multisensor covariance intersection fuser and three weighting fusers. *Inf. Fusion* **2013**, *14*, 177–185. [CrossRef]
23. Chen, L.; Arambel, P.O.; Mehra, R.K. Estimation under unknown correlation: Covariance intersection revisited. *IEEE Trans. Autom. Control.* **2002**, *47*, 1879–1882. [CrossRef]
24. Carlson, N.A. Federated square root filter for decentralized parallel processors. *IEEE Trans. Aerosp. Electron. Syst.* **1990**, *26*, 517–525. [CrossRef]
25. Lennart, L. *System Identification, Theory for the User*, 2nd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 1999.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

A Reliability-Based Method to Sensor Data Fusion

Wen Jiang *, Miaoyan Zhuang and Chunhe Xie

School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China; zhuang-my@mail.nwpu.edu.cn (M.Z.); xiechunhe@mail.nwpu.edu.cn (C.X.)

* Correspondence: jiangwen@nwpu.edu.cn or jiangwenpaper@hotmail.com; Tel.: +86-29-8843-1267

Received: 21 May 2017; Accepted: 3 July 2017; Published: 5 July 2017

Abstract: Multi-sensor data fusion technology based on Dempster–Shafer evidence theory is widely applied in many fields. However, how to determine basic belief assignment (BBA) is still an open issue. The existing BBA methods pay more attention to the uncertainty of information, but do not simultaneously consider the reliability of information sources. Real-world information is not only uncertain, but also partially reliable. Thus, uncertainty and partial reliability are strongly associated with each other. To take into account this fact, a new method to represent BBAs along with their associated reliabilities is proposed in this paper, which is named reliability-based BBA. Several examples are carried out to show the validity of the proposed method.

Keywords: sensor data fusion; Dempster–Shafer evidence theory; Gaussian distribution; reliability-based BBA

1. Introduction

In practical applications, there are various interferences in the working environment. Sensor data fusion technology can combine the related information from multiple sensors to enhance the robustness and safety of a system [1,2]. Hence, this technique has received significant attention in many fields, such as target tracking and recognition [3,4], complex network [5–7] and image processing [8–10]. Besides, information gathered from sensors is usually uncertain due to the change of the environment. It can degrade the performance of the information fusion system. Thus, how to handle uncertain information is a vital issue in the sensor fusion system. To solve this problem, many theories are presented by domestic and foreign scholars. Fuzzy sets theory was first introduced by Zadeh [11] in 1965 as an extension of the classical notion of set. It can be used in a wide range of domains in which information is incomplete or imprecise [12–14]. Dempster–Shafer evidence theory (evidence theory) acts as the pioneer in data fusion algorithms, which was proposed by Dempster [15] and extended by Shafer [16] subsequently. It is capable of managing epistemic and aleatoric uncertainty due to its framework. Possibility theory was introduced in 1978 by Zadeh [17]. It describes reasonably the meaning of information, especially the meaning of incomplete information within a possibilistic framework [18–21], which could be seen as the theory interconnecting fuzzy sets and evidence theory.

Within these theories, evidence theory has a good performance to process the uncertain information without the prior probability, which contributes to its wide application [22,23]. However, counter-intuitive results may be obtained when dealing with highly conflicting evidence. A famous example was illustrated by Zadeh [24]. Since then, many methods have been proposed to address this issue [25–27]. Martin et al. [28] proposed a conflict measures of a group of experts based on the distance of basic belief assignments. Smarandache et al. [29] presented a new normalization of a measure called contradiction to characterize the degree of discord or conflict inside a body of evidence. Martin [30] defined a conflict measure to quantify how the focal elements of two mass functions are included together. Deng et al. [31] considered a biological and evolutionary perspective to study the combination of evidences. Another open issue in evidence theory is how to determine

the basic belief assignment (BBA). So far, many methods to generate BBA have been put forward. Deneux [32] determined BBA by minimizing the mean squared differences between the classifier outputs and target values. Xu et al. [33] calculated an interval BBA from this matching degree by the modified Latin hypercube sampling Monte Carlo technique. Tabassian et al. [34] determined BBA in which the class memberships of training data are subject to ambiguity. A BBA method based on probability families encoded by possibility distributions and belief functions is presented by Baudrit and Dubois [35]. Mönks et al. [36,37] defined a fuzzy basic belief assignment (μ BBA) based on α -cut of fuzzy membership.

Comparing with the μ BBA, the idea of the proposed BBA determination method is similar to the μ BBA method to some extent. We both use the fuzzy membership function to obtain the degree of membership to the respective propositions, and the degree of membership is applied as BBA. However, the proposed method is still very different from the μ BBA method in some aspects. First, the BBA of the μ BBA method is defined in the real line. Namely, the determination of BBA is the determination of the possibility of sensor signal belonging to each interval proposition, which is the determination of value ranges. The BBA of the proposed method is defined in categorical data. Namely, the determination of the sensor signal is the determination of the category. Besides, in the proposed method, the fuzzy membership function is used to model the feature of each category proposition. However, in the μ BBA method, the fuzzy membership function is used to represent the knowledge of possibility, and the proposition is modeled by an interval. Hence, the proposed method and the μ BBA method are different. They are applicable to the different application backgrounds.

In reality, information is often not just uncertain, but also partially reliable. If we only consider one of them, then the whole complexity of real-world information cannot completely be covered, which may cause the incorrect fusion results. Hence, reliability evaluation is indispensable in practical applications [38,39]. Guo et al. [40] presented a new framework for sensor reliability evaluation in classification problems based on evidence theory. Yuan et al. [41] took the static reliability and dynamic reliability into consideration to handle the conflicting evidence. However, the reliability of these methods is measured from the support degree (consistency) among BBA. Namely, the reliability is obtained based on the given BBA, rather than from the information sources, which may lose part of the source information. Glock et al. [42] used the concepts of majority observation and consistency to monitor the sensor reliability based on the possibilistic framework. Ehlenbröcker et al. [43] proposed a method to generate a consistency-based reliability assessment for sensors, which is utilized to detect sensor defects based on groups of sensors instead combining all sensors at once. In the two reliability methods, the whole measure of reliability is time dependent. Besides, the static reliability is a prior reliability based on expert knowledge, and the dynamic reliability considers its former reliability and the consistency of observations. Comparing with that, the measure of reliability is not time dependent in this paper. The reliability is obtained based on the measure of sensor capability to distinguish the different targets. The capability to discriminate the difference classes is large under a certain attribute, and the reliability of the generated BBA is larger under this attribute.

The existing BBA methods pay more attention to the uncertainty of information, but do not simultaneously consider the reliability of information sources. In fuzzy sets theory, Zadeh [44] proposed the concept of the Z-number in 2011, which is an ordered pair of fuzzy numbers denoted by $Z = (A, B)$. The first component A is a fuzzy measurement of the uncertainty. The second component B is a measurement of the reliability of A . The Z-number can simultaneously describe the reliability of information sources. Originating from the idea of the Z-number, a new method to represent BBAs along with their associated reliability is proposed in this paper, which is named reliability-based BBA. Reliability-based BBA (BBA, R) is an ordered pair; its first component BBA is a mass function, and the second component R is a measurement of the reliability of the first component BBA .

The rest of this paper is organized as follows. The relevant concepts of evidence theory and pignistic probability are briefly recalled in Section 2. In Section 3, a reliability-based BBA is proposed. In Section 4, this method is compared with other methods by several examples. The conclusion is presented in Section 5.

2. Preliminaries

In this section, the relevant concepts of evidence theory and pignistic probability are briefly recalled.

2.1. Dempster–Shafer Evidence Theory

Evidence theory was introduced by Dempster [15] and then developed by Shafer [16]. It includes the following concepts: frame of discernment, mass function and Dempster’s combination rule, etc. These concepts contribute to its good performance in handling the uncertainty information [45–47].

2.1.1. Frame of Discernment

Let Θ be a set of N mutually-exclusive and collectively-exhaustive hypotheses, defined as:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_N\} \tag{1}$$

where Θ is called a frame of discernment. The power set of Θ is composed with 2^Θ , namely:

$$2^\Theta = \{\emptyset, \{\theta_1\}, \dots, \{\theta_N\}, \{\theta_1, \theta_2\}, \dots, \{\theta_1, \theta_2, \dots, \theta_i\}, \dots, \Theta\} \tag{2}$$

where \emptyset is denoted as the empty set. The N subsets containing only one element each are called the singleton subset proposition; the subsets containing more than one element each are called the compound subset proposition.

2.1.2. Mass Function

A mass function m is a mapping from 2^Θ to $[0, 1]$, formally defined as:

$$m : 2^\Theta \rightarrow [0, 1], \tag{3}$$

which satisfies the following conditions [16]:

$$\begin{cases} \sum_{A \subset 2^\Theta} m(A) = 1 \\ m(\emptyset) = 0. \end{cases} \tag{4}$$

The mass function m is also called the BBA function. Any subset A of 2^Θ , such that $m(A) > 0$, is called a focal element.

2.1.3. Dempster’s Combination Rule

Suppose m_1 and m_2 are two mass functions in the same frame of discernment. Dempster’s combination rule, which is denoted as $m = m_1 \oplus m_2$, is defined as follows [15]:

$$m(A) = \begin{cases} \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - k}, & A \neq \emptyset \\ 0, & A = \emptyset \end{cases} \tag{5}$$

where:

$$k = \sum_{B \cap C = \emptyset} m_1(B)m_2(C). \tag{6}$$

Here, k is regarded as a measure of conflict between m_1 and m_2 . The value of k is larger, and the conflict between the evidence is larger.

2.1.4. Discounting

Assuming that a BBA has a support degree of α , where $0 \leq \alpha \leq 1$, then this BBA is discounted by the following discounting rule [16]:

$$\begin{cases} m^\alpha(A) = \alpha \times m(A) \quad \forall A \subset 2^\Theta \\ m^\alpha(\Theta) = \alpha \times m(\Theta) + (1 - \alpha) \end{cases} \quad (7)$$

where A is any subset of the power set of the frame of discernment Θ .

2.2. Pignistic Probability

The pignistic probability function is introduced by Smets and Kennes [48] for decision making. Its procedure corresponds to the insufficient reason principle: if you need to build a probability distribution on n elements, given a lack of information, give a probability $1/n$ to each element. This procedure is repeated for each mass m . Let BetP be the pignistic probability distribution so derived. For all propositions $A \in \Theta$,

$$BetP(A) = \sum_{A \subseteq B \subseteq 2^\Theta} \frac{1}{|B|} \cdot \frac{m(B)}{1 - m(\emptyset)} \quad (8)$$

where \emptyset is denoted as the empty set. B is the proposition in mass function m , and $|B|$ is the cardinality of B .

3. The Proposed Method

In this section, the method of generating reliability-based BBA is given in detail. As shown in Figure 1, the method is expounded from five parts. In the first part, the models of training samples are built using the Gaussian membership functions. In the second part, according to the matching degree between the test sample and the attribute model, the first component BBA is generated, which is based on a previous work in the literature [49]. In the third part, the second component R (reliability of BBA) is measured, where both the similarity among classes under a certain attribute (static state) and the distance between the test sample and the models (dynamic state) are taken into account. The main contribution of this paper focuses on this part. Then, a reliability-based BBA (BBA, R) is obtained. Each (BBA, R) is discounted by the obtained reliability. Finally, these reliability-based BBAs are fused based on Dempster's combination rule. At the stage of modeling of sensor information, the uncertainty and the reliability of the information source are simultaneously considered in the proposed method, which can obtain a more adequate construction for the description of real-world information.

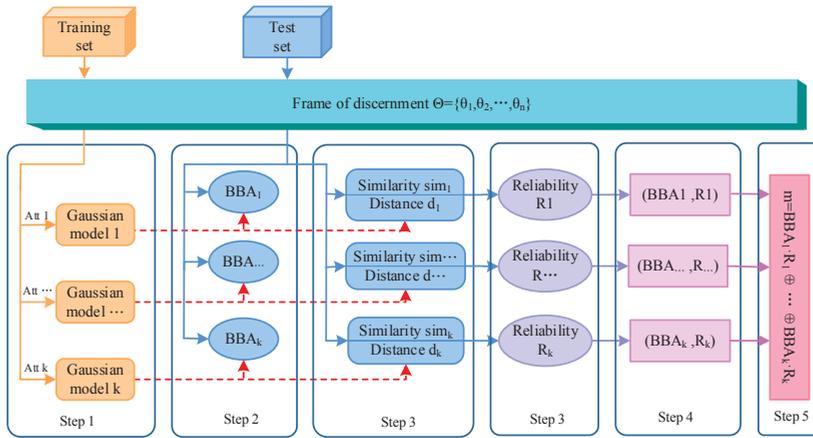


Figure 1. Flowchart of the proposed method.

3.1. The Modeling of Each Attribute

Due to the change of environment, the sensor data have usually a certain degree of fuzziness. In this case, the membership function can be used to represent the sample feature. Besides, there are some interruptions in the working process of the sensor, such as the mechanical noise and electromagnetic waves. In this case, the probability density function of the measured value of the same physical quantity is generally regarded as a form of Gaussian distribution. The Gaussian distribution possesses the following advantages [50]: first, if the error can be seen as the superposition of many independent random variables, then the error is supposed to have the form of the Gaussian distribution based on the central limit theorem. Second, many of the probability distributions of random variables in production and scientific experiments can be approximately described by the Gaussian distribution. Hence, the modeling of the training samples is built based on the Gaussian membership function in this paper.

Assuming that X is a sample space of the training set, then the Gaussian membership function of each attribute is defined as follows:

$$\mu(x) : X \rightarrow [0, 1], x \in X.$$

$\mu(x)$ can be gained by the following steps:

1. Suppose that there are n classes, namely the frame of discernment $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$. Each class θ_i ($i = 1, 2, \dots, n$) has k attributes.
2. For the training samples of class θ_i in the j -th attribute, the mean value \bar{X}_{ij} and the standard deviation σ_{ij} are calculated respectively as follows:

$$\begin{aligned} \bar{X}_{ij} &= \frac{1}{N} \sum_{l=1}^N x_{ijl}, \\ \sigma_{ij} &= \sqrt{\frac{1}{N-1} \sum_{l=1}^N (x_{ijl} - \bar{X}_{ij})^2}, \end{aligned} \tag{9}$$

where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$. N is the training sample size of class θ_i . x_{ijl} is the attribute value of the j -th attribute from the l -th training sample in class θ_i .

3. The Gaussian membership function of the j -th attribute of class θ_i is generated as follows:

$$\mu_i^j(x) = \exp\left[-\frac{(x - \bar{X}_{ij})^2}{2\sigma_{ij}^2}\right] \tag{10}$$

where $-3\sigma_{ij} \leq x \leq 3\sigma_{ij}$, $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$.

3.2. Reliability-Based BBA Generation Method

In this section, a reliability-based BBA is introduced from two parts: one is the determination of BBA; another is the reliability measurement of BBA.

3.2.1. The Determination of BBA

In this section, a nested structure BBA function is introduced. As described in Figure 2, a singleton subset proposition is modeled by a Gaussian membership function. For example, the singleton subset proposition $\{A\}$ is produced by the membership function $\mu_A(x)$. The proposition with two elements is represented by the intersection of two singleton subset propositions. For instance, the compound subset proposition with two elements $\{AB\}$ can be constructed as follows:

$$\mu_{AB}(x) = \min(\mu_A(x), \mu_B(x)). \tag{11}$$

Namely, $\{AB\}$ is defined by the intersection $\mu_{AB}(x)$ of $\{A\}$ with $\{B\}$. Further, a compound subset proposition with three elements can be constructed by the intersection among three singleton subset propositions. For example, $\{ABC\}$ can be constructed as follows:

$$\mu_{ABC}(x) = \min(\mu_A(x), \mu_B(x), \mu_C(x)). \tag{12}$$

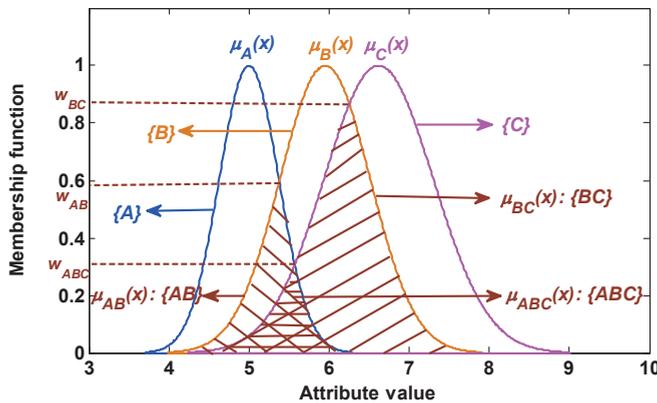


Figure 2. The modeling of the singleton subset and compound subset.

Suppose that $G(G \subset 2^\Theta)$ is a proposition and t is the feature information of a test sample under a certain attribute. The matching degree between t and G implies the plausibility of this test sample belonging to this proposition, which is defined as follows:

$$Pl(G|t) = \mu_G(x) |_{x=t} \tag{13}$$

where G can be a singleton subset proposition or a compound subset proposition. Equation (13) indicates that the plausibility is determined by the intersection between functions $\mu_G(x)$ and $x = t$.

That is, the plausibility of a test sample belonging to these propositions $\{A\}$, $\{AB\}$ and $\{ABC\}$ is denoted respectively as follows:

$$\begin{aligned}
 Pl(A|t) &= \mu_A(x) |_{x=t}, \\
 Pl(AB|t) &= \mu_{AB}(x) |_{x=t}, \\
 Pl(ABC|t) &= \mu_{ABC}(x) |_{x=t}.
 \end{aligned}
 \tag{14}$$

Then, the plausibility function, which measures the matching degree between the test sample and class proposition, is used to determinate BBA in this paper. Note that when the plausibility of a singleton subset proposition is equal to the plausibility of a compound subset proposition, this plausibility is only assigned to the compound subset proposition as its BBA. For example, as shown in Figure 3, for the test sample x , the plausibility of this test sample belonging to each proposition can be given as:

$$\begin{aligned}
 Pl(B|t) &= \mu_B(x) |_{x=7} = p_2, \\
 Pl(C|t) &= \mu_C(x) |_{x=7} = p_1, \\
 Pl(BC|t) &= \mu_{BC}(x) |_{x=7} = p_2, \\
 Pl(A|t) &= Pl(AB|t) = Pl(AC|t) = Pl(ABC|t) = 0
 \end{aligned}$$

Then, the BBA of each proposition is obtained as follows:

$$\begin{aligned}
 m(C) &= Pl(C|t) = p_1, \\
 m(BC) &= Pl(BC|t) = p_2, \\
 m(A) &= m(B) = m(AB) = m(AC) = m(ABC) = 0
 \end{aligned}$$

where since $Pl(B|t) = Pl(BC|t) = p_2$, the plausibility p_2 is assigned to the compound subset proposition BC , namely $m(BC) = p_2$, but $m(B) = 0$.

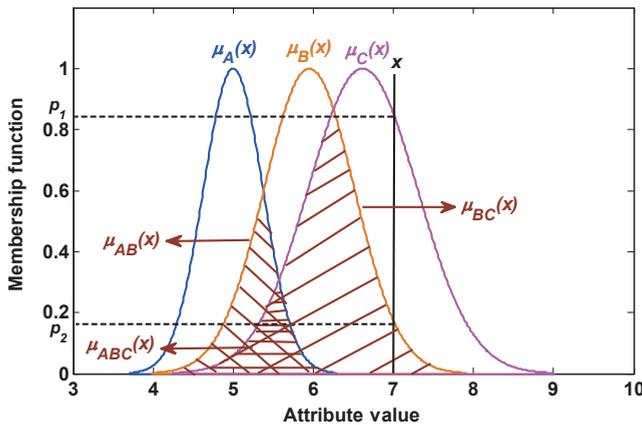


Figure 3. The determination of BBA.

Considering that the cumulative sum of the above gained BBAs may not be equal to one, the following rules are given to normalize BBAs: if the total sum is greater than one, these BBAs are normalized; if the total sum is less than one, the redundancy belief value $1 - \sum BBA$ is assigned to the universal set Θ , namely that is assigned to the unknown.

3.2.2. The Measurement of the Reliability of BBA

As shown in Figure 4, there exist two kinds of potential possibility of false classification: one is that class A is incorrectly recognized as class B, which is defined as $P(B | A)$; the other is that class B is incorrectly recognized as class A, which is defined as $P(A | B)$. The potential possibility leads to the generated BBA being not wholly reliable. This issue is not considered in the existing BBA methods. To address this issue, a method of measuring the BBA reliability is proposed in this paper.

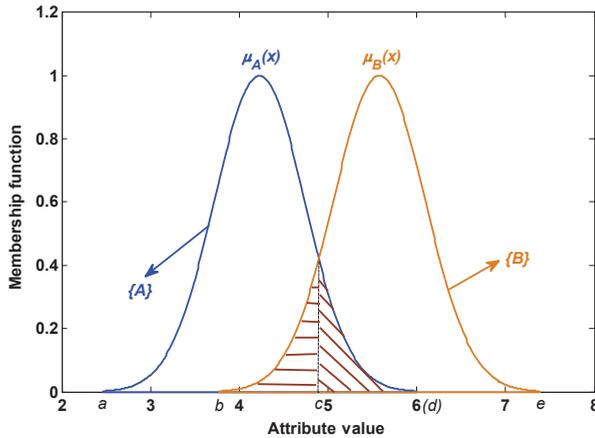


Figure 4. The overlapping area of the static attribute model.

1. The static reliability index based on the attribute model:

Assume there are two kinds of class A and B; from Figure 4, we can know that if a test sample comes from class A and its value locates in the range from c to d, the possibility of the false BBA $P(B | A) = \int_c^d \mu_A(x)dx / \int_a^d \mu_A(x)dx$. If a test sample comes from class B and its value locates in the range from b to c, the possibility of the false BBA $P(A | B) = \int_b^c \mu_B(x)dx / \int_b^e \mu_B(x)dx$. Hence, the total error rate of BBA is $P = P(B | A) + P(A | B)$. Namely, under a certain model ($\int_a^d \mu_A(x)dx$ and $\int_b^e \mu_B(x)dx$ are invariable), if the overlapping area $\int_c^d \mu_A(x)dx$ or $\int_b^c \mu_B(x)dx$ is larger, the similarity among classes is larger, then the possibility of generating the false BBA is larger, and vice versa. As analyzed above, a method of measuring the reliability of BBA is proposed based on the similarity among the attribute model, which is detailed as follows.

In the attribute $j(j = 1, 2, \dots, k)$, the similarity between Classes 1 and 2 is defined as:

$$sim_{12}^j = sim(\mu_1^j(x), \mu_2^j(x)) = \frac{\int \mu_{12}^j(x)dx}{\int \mu_1^j(x)dx + \int \mu_2^j(x)dx - \int \mu_{12}^j(x)dx} \tag{15}$$

where $\mu_1^j(x)$ and $\mu_2^j(x)$ are the membership functions of the j -th attribute of Class 1 and Class 2, respectively. $\mu_{12}^j(x)$ is the j -th attribute's membership function of the overlapping area between Class 1 and Class 2, namely the membership function of the compound subset $\{12\}$.

Then, a similarity matrix of the j -th attribute, SM_j , is obtained as follows:

$$SM_j = \begin{matrix} & \mu_1^j(x) & \mu_2^j(x) & \cdots & \mu_n^j(x) \\ \begin{matrix} \mu_1^j(x) \\ \mu_2^j(x) \\ \vdots \\ \mu_n^j(x) \end{matrix} & \begin{bmatrix} 1 & sim_{12}^j & \cdots & sim_{1n}^j \\ sim_{21}^j & 1 & \cdots & sim_{2n}^j \\ \vdots & \vdots & \ddots & \vdots \\ sim_{n1}^j & sim_{n2}^j & \cdots & 1 \end{bmatrix} \end{matrix} \tag{16}$$

where $sim_{il}^j(i, l = 1, 2, \dots, n)$ is the similarity between class i and class l in the j -th attribute.

Finally, based on the static model of the j -th attribute, the reliability of BBA generated from the j -th attribute is denoted as follows:

$$R_j^s = \sum_{i < l} (1 - sim_{il}^j) \tag{17}$$

where $j = 1, 2, \dots, k$ and $sim_{il}^j(i, l = 1, 2, \dots, n)$ is the similarity between class i and class l in the j -th attribute. Obviously, R_j^s implies that the larger the similarity, the lower the reliability of BBA generated from this attribute model.

2. The dynamic reliability index based on the test samples:

The static reliability index shows that the overlapping area among classes is larger, the similarity is larger, the possibility of generating the false BBA is larger, then the reliability of the generated BBA is smaller. This index can reflect that the reliability of BBA is affected by some static factors, such as the similarity among classes. However, the reliability of BBA is also affected by the test samples. For example, as shown in Figure 5a, although classes B and C have a large overlapping area, it is almost impossible that the test sample x_1 from class A is incorrectly classified as class B or C . In this case, the overlap degree between class B and C has little negative influence on the reliability of the generated BBA; whereas, in Figure 5b, the test sample x_2 from class C is easily incorrectly classified as class B . In this case, the overlap degree between classes B and C has a large negative influence on the reliability of the generated BBA. As analyzed above, the reliability of BBA is related to the test samples. To reflect this influence, a risk distance d is introduced in this section. It is denoted as the distance between the test sample and the overlapping area among attribute models. If the distance is larger, the influence of the overlapping area is smaller, the risk of the incorrect classification is smaller, then the reliability of the generated BBA is larger, and vice versa. The risk distance is produced as follows.

In the attribute $j(j = 1, 2, \dots, k)$, p_{12}^j is the maximum of the intersection between Class 1 and Class 2, namely the vertex of the overlapping area between Class 1 and Class 2. In this section, p_{12}^j is taken as the reference point of this overlapping area. Hence, the distance between the test sample x^j and the reference point p_{12}^j can be calculated to represent the risk distance. This reference point is defined as:

$$p_{12}^j = \sup \min(\mu_1(x)^j, \mu_2(x)^j). \tag{18}$$

where $\mu_1(x)^j$ and $\mu_2(x)^j$ are the Gaussian membership functions of Class 1 and Class 2 in the j -th attribute, respectively. Then, a vector P_j containing all reference points of the overlapping area among class $i(i = 1, 2, \dots, n)$ in the j -th attribute is obtained as follows:

$$P_j = [p_{12}^j \cdots p_{1n}^j \ p_{23}^j \cdots p_{2n}^j \cdots p_{(n-1)n}^j] \tag{19}$$

In the j -th attribute, the risk distance between the test sample x^j and the reference points $p_{(n-1)n}^j$ of the overlapping area between the classes $\mu_{n-1}^j(x)$ and $\mu_n^j(x)$ is formulated as:

$$d_{(n-1)n}^j = d(x^j, p_{(n-1)n}^j) = \frac{|x^j - p_{(n-1)n}^j|}{D_j} \tag{20}$$

where D_j represents the maximal interval comprised by all classes. Then, distance vector D_j^* of the j -th attribute is given as follows:

$$D_j^* = [d_{12}^j \ \cdots \ d_{1n}^j \ d_{23}^j \ \cdots \ d_{2n}^j \ \cdots \ d_{(n-1)n}^j] \tag{21}$$

Finally, the dynamic reliability index based on the test samples of the j -th attribute is denoted as follows:

$$R_j^d = e^{\sum_{l=2}^n d_{(l-1)l}^j} \tag{22}$$

where $d_{(l-1)l}^j$ is the distance between the test sample x^j and the reference points $p_{(l-1)l}^j$ of the overlapping area between class $\mu_{l-1}^j(x)$ and $\mu_l^j(x)$ in the j -th attribute. Obviously, R_j^d implies that the larger the risk distance, the larger the reliability of BBA generated from this attribute.

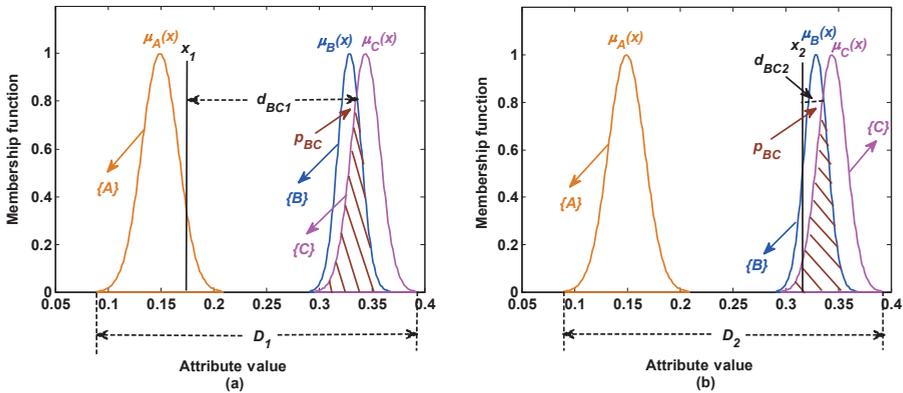


Figure 5. Reliability measure based on the dynamic test samples.

3. Comprehensive reliability measure:

Based on the above analysis, a method of measuring the BBA reliability is proposed. This method, which considers the comprehensive reliability based on the attribute model and the test samples, is more reasonable and effective. It is denoted as follows:

$$R_j = R_j^s \times R_j^d \tag{23}$$

where R_j^s and R_j^d are separately the static reliability index and the dynamic reliability index in the j -th attribute, which can be gained using Equations (17) and (22), respectively.

Suppose that there are K attributes of each class, the reliability of the generated BBA in the j -th attribute are normalized as:

$$R_j^s = \frac{R_j}{\max(R_k)} \tag{24}$$

where $k = 1, 2, \dots, K$.

According to Section 3.2.1 and Section 3.2.2, both the first component *BBA* and the second component *R* of reliability-based BBA can be acquired. Finally, the proposed reliability-based BBA (*BBA, R*) is obtained.

3.3. Sensor Data Fusion

In this paper, the reliability-based BBA is first translated into the classical BBA based on the discounting method [16]. After that, Dempster’s combination rule can be used to fuse these BBAs.

Assuming that in the frame of discernment $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$, there are *k* reliability-based BBAs (*BBA_j, R_j*), (*j* = 1, 2, …, *k*). Based on the discounting method, the BBAs generated from sensors are discounted as follows:

$$\begin{cases} m_j^R(A) = R_j \times m_j(A) \quad \forall A \subset 2^\Theta \\ m_j^R(\Theta) = R_j \times m_j(\Theta) + (1 - R_j) \end{cases} \quad (25)$$

where *R_j* is the reliability of *BBA_j* of the *j*-th attribute.

Eventually, these reliability-based BBAs are fused using Dempster’s combination rule. The maximum pignistic probability is taken as the decision-making criterion in this paper. Hence, the final mass function is transformed to pignistic probability, and the final decision-making can be done.

4. Numerical Example

To evaluate the validity of the reliability-based BBA, several experiments of two datasets and a fault diagnosis are performed in this section.

4.1. Experiments on Two Datasets: Five-Fold Cross-Validation

In this section, two kinds of datasets, including Iris and Wine, are selected from the UCI databases (UCI Machine Learning Repository: <http://archive.ics.uci.edu/ml/datasets.html>.) to evaluate the proposed method. Within these, the Iris dataset contains three classes of 50 samples each, and each class has four attributes, which is the well-known database in pattern recognition. The Wine dataset contains three classes, and each class has 13 attributes. First of all, the proposed method is compared with the method, which uses the same BBA method, but does not consider the reliability of BBA. The comparison results of five-fold cross-validation are shown in Figure 6.

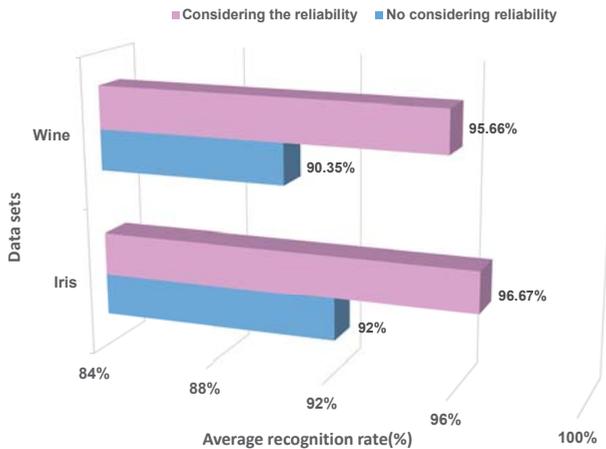


Figure 6. The comparison of considering the reliability or not.

The above results show that after considering the reliability of BBA, the recognition rates of two datasets are all increased to some extent. Namely, the proposed method that measures the reliability of BBA is valid and reasonable.

To further evaluate this method, a comparison experiment between our method and other classifiers is carried out. Here, we consider the following three kinds of classifiers: support vector machine with radial basis function (SVM-RBF), decision tree (REPTree) and naive Bayesian (NB). Within these, SVM and NB are both the top ten data-mining algorithms [51]. REPTree is also a well-known machine-learning algorithm. The comparison results of five-fold cross-validation are shown in Figure 7.

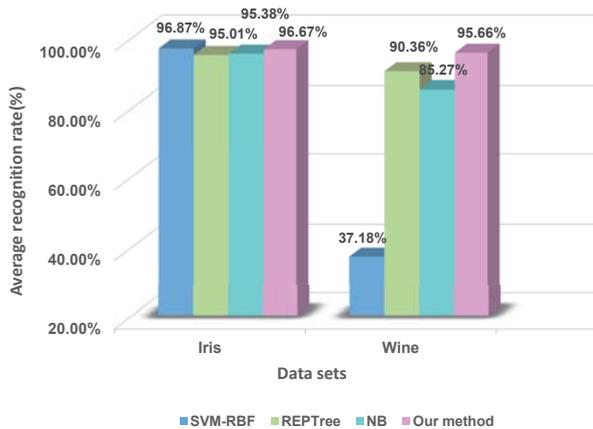


Figure 7. The comparing results between our method and other classifiers.

From the above experimental results, it can be found that: in the Iris dataset, the recognition rates of the proposed method and the classical classifiers are greater than 90%, namely they are all effective in this dataset. In the Wine dataset, the recognition rate of our method is 95.66%; NB is 85.27%; REPTree is 90.36%; and SVM-RBF is 37.18%. This shows that the proposed method has competitive performances contrasting with these selected classifiers. What is more, the reliability of BBA is measured at the stage of BBA generation, which is more reasonable.

4.2. An Application Example of Fault Diagnosis

To evaluate the validity of this method in engineering applications, a case study of the fault diagnosis of motor rotor is executed. There are three kinds of fault: $F_1 = \{Rotorunbalance\}$, $F_2 = \{Rotormisalignment\}$ and $F_3 = \{Pedestallooseness\}$. Three vibration acceleration sensors and a vibration displacement sensor are placed in different installation positions to collect the vibration signal. Vibration displacement and acceleration vibration frequency amplitudes at the frequencies of $1\times$, $2\times$ and $3\times$ are taken as the fault feature variables. The relevant data are acquired from the literature [52], which is cited in the Appendix. The method is compared with the method that does not consider the reliability of BBA and other classifiers, respectively. The comparison results of five-fold cross-validation are shown in Table 1.

Table 1. The comparing the results of fault diagnosis.

Methods	Classes			Overall Average
	F_1	F_2	F_3	
Not considering reliability	99.00%	95.50%	100%	98.17%
Support vector machine	94.15%	92.86%	100%	95.67%
Decision tree	99.05%	98.68%	99.78%	99.17%
Naive Bayesian	98.05%	96.94%	100%	98.33%
The proposed method	100%	100%	100%	100%

The above experiments can evaluate the rationality and the effectiveness of the presented method well. The advantages of our method are concluded as follows:

- Based on the idea of the Z-number, an ordered pair (BBA, R) is proposed to represent BBA along with its associated reliability. The first component BBA is a mass function; the second component R is a measurement of the reliability of the first component. According to this ordered pair, the reliability of BBA can be measured well at the stage of BBA generation.
- In the process of measuring the reliability of BBA, the information about two things is taken into account. One is the similarity among classes (static information). Another is the risk distance between the test samples and the overlapping area among classes (dynamic information). This makes the results truer and more credible.
- The proposed method is based on a feasible method of measuring the reliability of BBA, which can be replaced with other measure methods for different applications. Namely, this method is flexible and easy to extend in many applications.

5. Conclusions

In the multi-sensors data fusion based on evidence theory, how to determine BBA is an open issue. In this paper, a novel method named the reliability-based BBA is proposed. Within this method, first, the models of training samples are built using the Gaussian membership functions. Second, the BBA of every test sample is generated based on the matching degree between the test sample and the attribute model. Then, the reliability of BBA is measured according to both the similarity among classes and the risk distance between the test samples and the overlapping area among classes. Finally, a reliability-based BBA can be generated. Several performed experiments verify that the proposed method is effective. In the future, we will try to apply the proposed method to more practical applications and research a new combination rule for reliability-based BBA.

Acknowledgments: The work is partially supported by the National Natural Science Foundation of China (Grant No. 61671384), the Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016JM6018), the Aviation Science Foundation (Program No. 20165553036), the Fundamental Research Funds for the Central Universities (Program No. 3102017OQD020) and the Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University (Program No. Z2017142).

Author Contributions: Wen Jiang and Miaoyan Zhuang designed the research. Wen Jiang performed the experiments. Chunhe Xie analyzed the data. Miaoyan Zhuang wrote the manuscript. All authors read and approved the final manuscript.

Conflicts of Interest: The authors declare that they have no competing interests.

Appendix A. Experimental Data of Fault Diagnosis

There are three kinds of faults in a motor rotor: $F = \{X, Y, Z\} = \{\text{Rotorunbalance}, \text{Rotormisalignment}, \text{Pedestallooseness}\}$, which has four fault feature variables. At the same time interval, each fault feature of each fault is continuously observed 40 times, which is taken as a group of observations. A total of five groups is measured in each fault feature of each fault. Hence, $i = 1, 2, 3, 4$ respectively represent four fault feature variables: displacement, acceleration

frequency of $1 \times$, $2 \times$ and $3 \times$. $j = 1, 2, 3, 4, 5$ respectively represent the group numbers of the measured data. Five groups of observations of each feature variables of each fault are shown in Table A1.

Table A1. Experimental data of fault diagnosis [52].

Groups		Observations								
X11	0.1663	0.1590	0.1568	0.1485	0.1723	0.2006	0.1903	0.1908	0.1986	0.1843
	0.1785	0.1610	0.1579	0.1511	0.1532	0.1647	0.1628	0.1646	0.1634	0.1642
	0.1648	0.1640	0.1674	0.0661	0.1659	0.1650	0.1633	0.1632	0.1604	0.1542
	0.1555	0.1562	0.1540	0.1564	0.1557	0.1542	0.1546	0.1571	0.1537	0.1536
X12	0.154	0.1518	0.1537	0.1548	0.1542	0.1538	0.1545	0.1537	0.1571	0.1560
	0.1584	0.1552	0.1586	0.1574	0.1569	0.1565	0.1551	0.1585	0.1585	0.1593
	0.1548	0.1558	0.1547	0.1593	0.1532	0.1632	0.1575	0.159	0.1594	0.1541
	0.165	0.1674	0.1651	0.1604	0.1787	0.1818	0.1820	0.1656	0.1658	0.1644
X13	0.1647	0.1647	0.1654	0.1651	0.1656	0.1653	0.1652	0.1652	0.1648	0.1649
	0.1653	0.1650	0.1650	0.1652	0.1653	0.1652	0.1648	0.1647	0.1646	0.1645
	0.1651	0.1652	0.1652	0.1649	0.1650	0.1643	0.1640	0.1639	0.1641	0.1633
	0.1632	0.1629	0.1630	0.1630	0.1634	0.1631	0.1634	0.1629	0.1632	0.1629
X14	0.1630	0.1629	0.1627	0.1626	0.1622	0.1624	0.1627	0.1618	0.1614	0.1617
	0.1621	0.1615	0.1618	0.1611	0.1614	0.1610	0.1612	0.1611	0.1616	0.1612
	0.1612	0.1613	0.1623	0.1616	0.1621	0.1613	0.1611	0.1610	0.1610	0.1613
	0.1615	0.1616	0.1618	0.1616	0.1614	0.1612	0.1606	0.1614	0.1619	0.1614
X15	0.1609	0.1610	0.1612	0.1615	0.1609	0.1606	0.1604	0.1606	0.1605	0.1601
	0.1604	0.1608	0.1610	0.1603	0.1599	0.1601	0.1602	0.1599	0.1598	0.1598
	0.1598	0.1596	0.1597	0.1595	0.1593	0.1577	0.1580	0.1576	0.1577	0.1579
	0.1598	0.1596	0.1597	0.1595	0.1593	0.1577	0.1580	0.1576	0.1577	0.1579
X21	0.1612	0.1620	0.1612	0.1610	0.1385	0.1222	0.1475	0.1306	0.1210	0.1501
	0.1548	0.1577	0.1622	0.1618	0.1621	0.1665	0.1639	0.1652	0.1625	0.1637
	0.1645	0.1645	0.1650	0.1649	0.1650	0.1630	0.1493	0.1533	0.1474	0.1460
	0.1489	0.1499	0.1495	0.1491	0.1489	0.1503	0.1507	0.1505	0.1477	0.1496
X22	0.1517	0.1496	0.1504	0.1498	0.1528	0.1519	0.1534	0.1516	0.1555	0.1520
	0.1512	0.1546	0.1538	0.1551	0.1563	0.1536	0.1543	0.1519	0.1514	0.1520
	0.1501	0.1514	0.1483	0.1499	0.1502	0.1550	0.1537	0.1507	0.1557	0.1537
	0.1556	0.1545	0.1529	0.1500	0.1380	0.1343	0.1346	0.1544	0.1458	0.1424
X23	0.1464	0.1460	0.1446	0.1448	0.1476	0.1464	0.1434	0.1432	0.1450	0.1420
	0.1448	0.1452	0.1456	0.1462	0.1464	0.1464	0.1444	0.1440	0.1422	0.1442
	0.1470	0.1478	0.1468	0.1482	0.1472	0.1462	0.1478	0.1494	0.1488	0.1496
	0.1480	0.1476	0.1502	0.1496	0.1488	0.1488	0.1484	0.1534	0.1490	0.1486
X24	0.1466	0.1460	0.1438	0.1458	0.1488	0.1466	0.1494	0.1502	0.1486	0.1488
	0.1512	0.1490	0.1470	0.1478	0.1484	0.1490	0.1474	0.1456	0.1464	0.1446
	0.3468	0.1484	0.1478	0.1486	0.1470	0.1448	0.1460	0.1458	0.1458	0.1456
	0.1452	0.1470	0.1470	0.1458	0.1450	0.1456	0.1462	0.1442	0.1464	0.1468
X25	0.1484	0.1474	0.1488	0.1460	0.1462	0.1464	0.1452	0.1450	0.1438	0.1434
	0.1438	0.1438	0.1436	0.1436	0.1432	0.1412	0.1428	0.1418	0.1422	0.1422
	0.1432	0.1406	0.1420	0.1402	0.1410	0.1418	0.1432	0.1450	0.1418	0.1424
	0.1412	0.1408	0.1412	0.1390	0.1412	0.1398	0.1406	0.1394	0.1392	0.1382
X31	0.1221	0.1219	0.1207	0.1215	0.1222	0.1296	0.1235	0.1295	0.1280	0.1233
	0.1218	0.1159	0.1163	0.1195	0.1190	0.1271	0.1247	0.1232	0.1233	0.1237
	0.1210	0.1227	0.1233	0.1222	0.1252	0.1230	0.1049	0.1033	0.0899	0.1003
	0.1044	0.1060	0.1064	0.1042	0.1072	0.1071	0.1070	0.1070	0.1041	0.1049
X32	0.1068	0.1063	0.1069	0.1057	0.1091	0.1061	0.1094	0.1067	0.1109	0.1111
	0.1112	0.1096	0.1074	0.1085	0.1109	0.1116	0.1110	0.1113	0.1106	0.1110
	0.1091	0.1080	0.1044	0.1098	0.1084	0.1102	0.1078	0.1087	0.1111	0.1116
	0.1124	0.1128	0.1110	0.1078	0.1101	0.1115	0.1131	0.1108	0.1111	0.1079

Table A1. Cont.

Groups	Observations									
X33	0.1105	0.1092	0.1074	0.1096	0.1055	0.1076	0.1003	0.1031	0.1040	0.1046
	0.1041	0.1021	0.1041	0.1053	0.1057	0.1038	0.1029	0.1037	0.1012	0.0997
	0.1020	0.1020	0.0990	0.1049	0.1066	0.1065	0.1098	0.1102	0.1076	0.1116
	0.1097	0.1150	0.1120	0.1078	0.1106	0.1075	0.1061	0.1090	0.1098	0.1105
X34	0.1105	0.1081	0.1075	0.1059	0.1097	0.1105	0.1086	0.1085	0.1095	0.1084
	0.1093	0.1113	0.1122	0.1139	0.1140	0.1129	0.1119	0.1107	0.1119	0.1137
	0.1128	0.1122	0.1104	0.1129	0.1130	0.1143	0.1132	0.1132	0.1115	0.1111
	0.1123	0.1124	0.1117	0.1120	0.1130	0.1127	0.1158	0.1145	0.1138	0.1144
X35	0.1160	0.1137	0.1159	0.1164	0.1158	0.1165	0.1167	0.1160	0.1155	0.1175
	0.1170	0.1175	0.1168	0.1191	0.1190	0.1191	0.1190	0.1211	0.1196	0.1187
	0.1191	0.1202	0.1200	0.1205	0.1194	0.1193	0.1195	0.1180	0.1190	0.1194
	0.1197	0.1194	0.1173	0.1187	0.1169	0.1179	0.1184	0.1197	0.1194	0.1196
X41	4.4090	4.3780	4.3430	4.2950	4.2890	4.2890	4.2740	4.1840	4.1820	4.2020
	4.2130	4.2240	4.2220	4.2250	4.2160	4.2220	4.2210	4.2410	4.2200	4.2180
	4.2260	4.2430	4.2390	4.2370	4.2270	4.2300	4.2210	4.2200	4.2430	4.6660
	4.4540	4.4370	4.4380	4.4410	4.4400	4.4350	4.4330	4.4430	4.4460	4.4420
X42	4.4480	4.4380	4.4420	4.4320	4.4270	4.4320	4.4220	4.4320	4.4240	4.4270
	4.4590	4.4240	4.4650	4.4180	4.4200	4.4180	4.4190	4.4230	4.4200	4.4460
	4.4210	4.4040	4.4120	4.4000	4.4100	4.4150	4.4070	4.4120	4.3920	4.4020
	4.3930	4.3920	4.3860	4.3890	4.3820	4.3790	4.4120	4.3750	4.3740	4.3790
X43	4.3680	4.3840	4.3800	4.3690	4.3840	4.3830	4.3830	4.3820	4.3830	4.3850
	4.3800	4.3800	4.3710	4.3720	4.3740	4.3890	4.3720	4.3670	4.3750	4.3650
	4.3600	4.3570	4.3640	4.3570	4.3550	0.3570	4.3480	4.3470	4.3470	4.3400
	4.3460	4.3360	4.3190	4.3300	4.3480	4.3500	4.3500	4.3460	4.3500	4.3500
X44	4.4000	4.3440	4.3410	4.3420	4.3510	4.3450	4.3370	4.3370	4.3340	4.3330
	4.3330	4.3210	4.3250	4.3180	4.3300	4.3100	4.3190	4.3160	4.3160	4.3150
	4.3090	4.3040	4.3060	4.3050	4.3010	4.3000	4.2960	4.2940	4.2860	4.2860
	4.2890	4.2940	4.2900	4.3070	4.2890	4.2800	4.2820	4.2880	4.2810	4.2980
X45	4.3000	4.2930	4.2980	4.3303	4.2990	4.2870	4.3030	4.2910	4.2950	4.3050
	4.3020	4.3120	4.3250	4.3090	4.3240	4.3210	4.3240	4.3230	4.3270	4.3290
	4.3230	4.3290	4.3290	4.3350	4.3210	4.3240	4.3270	4.4620	4.4120	4.3740
	4.3960	4.3730	4.3550	4.3540	4.3500	4.3430	4.3470	4.3550	4.3380	4.3310
Y11	0.1666	0.1666	0.1670	0.1696	0.1665	0.1671	0.1652	0.1663	0.1656	0.1656
	0.1659	0.1655	0.1640	0.1634	0.1631	0.1618	0.1617	0.1615	0.1597	0.1592
	0.1584	0.1585	0.1575	0.1578	0.1573	0.1567	0.1834	0.1825	0.1827	0.1822
	0.1828	0.1817	0.1820	0.1823	0.1808	0.1818	0.1814	0.1816	0.1808	0.1807
Y12	0.1809	0.1794	0.1799	0.1799	0.1788	0.1795	0.1785	0.1785	0.1780	0.1777
	0.1778	0.1777	0.1766	0.1767	0.1761	0.1770	0.1757	0.1765	0.1755	0.1755
	0.1746	0.1757	0.1741	0.1743	0.1741	0.1732	0.1736	0.1723	0.1730	0.1708
	0.1708	0.1702	0.1691	0.1686	0.1683	0.1676	0.1668	0.1670	0.1649	0.1644
Y13	0.1637	0.1639	0.1655	0.1641	0.1643	0.1641	0.1625	0.2038	0.2037	0.2033
	0.2014	0.2028	0.2022	0.2026	0.2014	0.2013	0.2007	0.2012	0.1999	0.2012
	0.1998	0.1999	0.1988	0.1992	0.1988	0.1985	0.1977	0.1976	0.1979	0.1981
	0.1966	0.1973	0.1979	0.1984	0.1973	0.1969	0.1963	0.1960	0.1953	0.1941
Y14	0.1958	0.1952	0.1954	0.1938	0.1940	0.1956	0.1945	0.1937	0.1954	0.1947
	0.1950	0.1955	0.1947	0.1956	0.1945	0.1932	0.1942	0.1925	0.1924	0.1934
	0.1904	0.1905	0.1909	0.1906	0.1989	0.1898	0.1891	0.1892	0.1886	0.1880
	0.1884	0.1890	0.1874	0.1872	0.1880	0.1855	0.1862	0.1866	0.1849	0.1841
Y15	0.1857	0.1844	0.1837	0.1831	0.1835	0.1826	0.1828	0.1834	0.1833	0.1837
	0.1822	0.1829	0.1823	0.1807	0.1833	0.1835	0.1832	0.1828	0.1816	0.1820
	0.1805	0.1808	0.1803	0.1795	0.1785	0.1794	0.1795	0.1788	0.1786	0.1781
	0.1771	0.1775	0.1774	0.1769	0.1780	0.1778	0.1758	0.1740	0.1736	0.1738

Table A1. Cont.

Groups		Observations									
Y21	0.3111	0.3124	0.3205	0.3268	0.3225	0.3268	0.3305	0.3245	0.3247	0.3245	
	0.3300	0.3279	0.3265	0.3221	0.3209	0.3227	0.3196	0.3150	0.3193	0.3182	
	0.3148	0.3122	0.3133	0.3107	0.3131	0.3071	0.3412	0.3401	0.3357	0.3466	
	0.3422	0.3390	0.3372	0.3364	0.3398	0.3392	0.3384	0.3383	0.3344	0.3394	
Y22	0.3386	0.3342	0.3364	0.3338	0.3381	0.3388	0.3347	0.3348	0.3321	0.3367	
	0.3367	0.3322	0.3300	0.3309	0.3346	0.3341	0.3335	0.3303	0.3320	0.3317	
	0.3295	0.3265	0.3299	0.3267	0.3271	0.3253	0.3297	0.3247	0.3243	0.3269	
	0.3229	0.3211	0.3171	0.3202	0.3170	0.3125	0.3144	0.3165	0.3079	0.3087	
Y23	0.3117	0.3095	0.3152	0.3222	0.3171	0.3169	0.3157	0.3480	0.3498	0.3469	
	0.3447	0.3476	0.3507	0.3470	0.3403	0.3359	0.3412	0.3399	0.3459	0.3449	
	0.3479	0.3422	0.3446	0.3471	0.3467	0.3461	0.3421	0.3413	0.3416	0.3457	
	0.3423	0.3439	0.3423	0.3465	0.3405	0.3399	0.3372	0.3387	0.3333	0.3349	
Y24	0.3419	0.3436	0.3510	0.3392	0.3354	0.3350	0.3500	0.3354	0.3358	0.3349	
	0.3385	0.3414	0.3351	0.3394	0.3371	0.3374	0.3370	0.3365	0.3342	0.3389	
	0.3386	0.3394	0.3374	0.3355	0.3357	0.3312	0.3274	0.3353	0.3351	0.3325	
	0.3305	0.3314	0.3304	0.3238	0.3315	0.3259	0.3253	0.3308	0.3215	0.3233	
Y25	0.3282	0.3208	0.3211	0.3138	0.3144	0.3199	0.3182	0.3196	0.3205	0.3180	
	0.3166	0.3170	0.3181	0.3139	0.3212	0.3254	0.3238	0.3193	0.3204	0.3168	
	0.3148	0.3204	0.3146	0.3132	0.3191	0.3164	0.3141	0.3165	0.3137	0.3160	
	0.3135	0.3137	0.3188	0.3177	0.3193	0.3239	0.3158	0.3236	0.3291	0.3262	
Y31	0.2517	0.2634	0.2590	0.2808	0.2869	0.2827	0.2913	0.2909	0.2893	0.2903	
	0.2999	0.2961	0.2930	0.3040	0.2971	0.3125	0.2968	0.2979	0.2998	0.3003	
	0.3023	0.2986	0.3008	0.3022	0.3017	0.3218	0.2338	0.2414	0.2510	0.2498	
	0.2424	0.2451	0.2477	0.2473	0.2494	0.2523	0.2523	0.2496	0.2557	0.2591	
Y32	0.2485	0.2534	0.2636	0.2670	0.2661	0.2641	0.2581	0.2637	0.2733	0.2735	
	0.2644	0.2622	0.2669	0.2713	0.2663	0.2720	0.2753	0.2758	0.2722	0.2755	
	0.2710	0.2870	0.2820	0.2770	0.2727	0.2761	0.2812	0.2777	0.2880	0.2919	
	0.2882	0.2784	0.2788	0.2792	0.2799	0.2731	0.2717	0.2851	0.2606	0.2696	
Y33	0.2786	0.2774	0.2921	0.2991	0.2982	0.2974	0.2980	0.2015	0.1872	0.1865	
	0.2016	0.1980	0.1982	0.2022	0.2071	0.2020	0.1882	0.1877	0.2065	0.2057	
	0.2052	0.2143	0.2135	0.2261	0.2110	0.2077	0.2089	0.2134	0.2161	0.2119	
	0.2109	0.2130	0.2180	0.2096	0.2102	0.2152	0.2137	0.2110	0.2113	0.2126	
Y34	0.2170	0.2130	0.2190	0.2192	0.2112	0.2214	0.2166	0.2137	0.2109	0.2024	
	0.2117	0.2102	0.2087	0.2050	0.2149	0.2134	0.2067	0.2140	0.2239	0.2153	
	0.2144	0.2103	0.2145	0.2190	0.2250	0.2137	0.2060	0.2153	0.2132	0.2160	
	0.2079	0.2047	0.2130	0.2058	0.2174	0.2138	0.2142	0.2138	0.2022	0.2169	
Y35	0.2206	0.2133	0.2141	0.2031	0.2073	0.2099	0.2066	0.2052	0.2172	0.2131	
	0.2140	0.2184	0.2152	0.2099	0.2258	0.2264	0.2273	0.2322	0.2204	0.2248	
	0.2242	0.2251	0.2222	0.2317	0.2193	0.2262	0.2255	0.2332	0.2299	0.2289	
	0.2305	0.2398	0.2401	0.2306	0.2365	0.2398	0.2439	0.2595	0.2529	0.2557	
Y41	5.3920	5.3260	5.3080	5.2620	5.2800	5.2460	5.1950	5.2280	5.1840	5.1820	
	5.1590	5.1310	5.0980	4.9840	5.0190	4.9340	4.9260	4.9500	4.9690	4.8960	
	4.7990	4.8330	4.8220	4.7450	4.7840	4.8260	4.8960	4.8920	4.9120	4.8390	
	4.8230	4.7960	4.8000	4.8180	4.8240	4.8310	4.8370	4.8720	4.8410	4.8410	
Y42	4.8610	4.8220	4.6890	4.7250	4.7070	4.7300	4.6980	4.6810	4.6620	4.7610	
	4.7460	4.6870	4.7120	4.7080	4.6910	4.5130	4.4670	4.5120	4.5410	4.3910	
	4.4220	4.5130	4.5950	4.5810	4.5420	4.5400	4.5160	4.5220	4.5180	4.5660	
	4.5380	4.5450	4.4510	4.4570	4.4810	4.4860	4.4940	4.4690	4.4180	4.4170	
Y43	4.3700	4.4000	4.3950	4.3840	4.3740	4.3800	4.3310	4.3230	4.3140	4.2870	
	4.2300	4.2440	4.2500	4.2200	4.2150	4.2540	4.2100	4.1980	4.2550	4.2210	
	4.2110	4.2000	4.1810	4.1790	4.1840	4.1570	4.1440	4.1600	4.1150	4.0940	
	4.1230	4.1280	5.2340	5.2320	5.2110	5.2210	5.2280	5.2060	5.1800	5.1890	

Table A1. Cont.

Groups	Observations									
Y44	5.1510	5.1240	5.1230	5.1220	5.0830	5.0600	5.0930	5.0750	5.0490	5.0520
	5.0150	5.0250	5.0750	5.0150	4.9010	4.9300	4.9080	4.8860	4.8780	4.9040
	4.8980	4.8830	4.8510	4.8510	4.8370	4.9340	8.8960	4.8160	4.7640	4.7940
	4.8010	4.7670	4.7450	4.7540	4.7710	4.7560	4.7540	4.7360	4.6780	4.6650
Y45	4.6770	4.6610	4.6500	4.6280	4.6440	4.6320	4.6120	4.4620	4.6770	4.6580
	4.6290	4.6220	4.6300	4.6140	4.6260	4.6130	4.5850	4.5690	4.5820	4.5500
	4.5330	4.5520	4.5040	4.4760	4.5660	4.5280	4.5550	4.5230	4.5190	4.5390
	4.5220	4.5210	4.5090	4.4870	4.5270	4.4730	4.4710	4.4900	4.4570	4.4510
Z11	0.3207	0.3213	0.3213	0.3235	0.3322	0.3419	0.3434	0.3440	0.3454	0.3461
	0.3474	0.3476	0.3432	0.3468	0.3439	0.3423	0.3440	0.3430	0.3436	0.3420
	0.3416	0.3402	0.3373	0.3403	0.3414	0.3423	0.3420	0.3423	0.3425	0.3379
	0.3379	0.3391	0.3386	0.3355	0.3352	0.3361	0.3333	0.3333	0.3315	0.3347
Z12	0.3347	0.3320	0.3323	0.3327	0.3329	0.3287	0.3304	0.3312	0.3285	0.3287
	0.3309	0.3270	0.3274	0.3285	0.3283	0.3305	0.3274	0.3261	0.3264	0.3251
	0.3271	0.3252	0.3275	0.3275	0.3287	0.3270	0.3269	0.3297	0.3266	0.3308
	0.3293	0.3304	0.3323	0.3305	0.3305	0.3330	0.3339	0.3342	0.3312	0.3315
Z13	0.3312	0.3301	0.3315	0.3307	0.3315	0.3320	0.3311	0.3327	0.3292	0.3301
	0.3315	0.3289	0.3246	0.3267	0.3295	0.3270	0.3238	0.3264	0.3251	0.3264
	0.3260	0.3247	0.3224	0.3235	0.3249	0.3230	0.3232	0.3273	0.3249	0.3270
	0.3218	0.3244	0.3006	0.3030	0.3041	0.3174	0.3220	0.3196	0.3241	0.3251
Z14	0.3263	0.3266	0.3282	0.3270	0.3290	0.3198	0.3237	0.3229	0.3261	0.3238
	0.3259	0.3221	0.3309	0.3271	0.3242	0.3235	0.3240	0.3261	0.3294	0.3287
	0.3267	0.3277	0.3263	0.3262	0.3278	0.3276	0.3271	0.3267	0.3289	0.3270
	0.3266	0.3299	0.3068	0.3148	0.3322	0.3323	0.3320	0.3336	0.3326	0.3322
Z15	0.3326	0.3317	0.3301	0.3316	0.3336	0.3280	0.3292	0.3297	0.3283	0.3283
	0.3264	0.3279	0.3275	0.3294	0.3245	0.3268	0.3261	0.3262	0.3253	0.3272
	0.3270	0.3252	0.3284	0.3253	0.3265	0.3277	0.3291	0.3287	0.3256	0.3239
	0.3248	0.3261	0.3252	0.3249	0.3254	0.3290	0.3275	0.3274	0.3274	0.3251
Z21	0.2893	0.2863	0.2801	0.2847	0.3271	0.3448	0.3409	0.3346	0.3249	0.3425
	0.3360	0.3368	0.3361	0.3411	0.3434	0.3459	0.3460	0.3481	0.3518	0.3495
	0.3478	0.3477	0.3506	0.3470	0.3470	0.3501	0.3477	0.3561	0.3489	0.3529
	0.3539	0.3544	0.3525	0.3515	0.3560	0.3596	0.3567	0.3616	0.3602	0.3589
Z22	0.3541	0.3561	0.3607	0.3636	0.3614	0.3595	0.3586	0.3575	0.3574	0.3563
	0.3601	0.3619	0.3647	0.3599	0.3621	0.3647	0.3557	0.3457	0.3558	0.3509
	0.3525	0.3527	0.3484	0.3452	0.3474	0.3438	0.3500	0.3447	0.3429	0.3508
	0.3397	0.3375	0.3503	0.3421	0.3421	0.3362	0.3328	0.3409	0.3391	0.3364
Z23	0.3287	0.3323	0.3313	0.3416	0.3315	0.3352	0.3396	0.3349	0.3402	0.3406
	0.3472	0.3526	0.3439	0.3462	0.3427	0.3492	0.3507	0.3550	0.3456	0.3522
	0.3480	0.3397	0.3474	0.3499	0.3503	0.3365	0.3450	0.3516	0.3506	0.3528
	0.3493	0.3546	0.2995	0.3094	0.2950	0.3479	0.3361	0.3394	0.3484	0.3441
Z24	0.3469	0.3380	0.3356	0.3378	0.3385	0.3338	0.3396	0.3345	0.3363	0.3426
	0.3333	0.3298	0.3335	0.3339	0.3397	0.3349	0.3357	0.3361	0.3401	0.3382
	0.3379	0.3356	0.3309	0.3333	0.3328	0.3330	0.3412	0.3334	0.3264	0.3297
	0.3302	0.3318	0.2961	0.3143	0.3616	0.3506	0.3463	0.3446	0.3412	0.3393
Z25	0.3454	0.3396	0.3453	0.3455	0.3517	0.3426	0.3590	0.3516	0.3481	0.3502
	0.3440	0.3428	0.3455	0.3404	0.3518	0.3517	0.3389	0.3481	0.3382	0.3530
	0.3471	0.3566	0.3554	0.3539	0.3576	0.3536	0.3480	0.3568	0.3567	0.3524
	0.3587	0.3578	0.3535	0.3602	0.3565	0.3490	0.3532	0.3541	0.3507	0.3467
Z31	0.1810	0.1864	0.1803	0.1829	0.1605	0.1441	0.1436	0.1412	0.1414	0.1476
	0.1502	0.1477	0.1507	0.1469	0.1490	0.1512	0.1461	0.1497	0.1511	0.1488
	0.1486	0.1480	0.1493	0.1451	0.1520	0.1537	0.1498	0.1478	0.1471	0.1496
	0.1467	0.1443	0.1446	0.1420	0.1454	0.1365	0.1347	0.1373	0.1380	0.1446

Table A1. Cont.

Groups	Observations									
Z32	0.1434	0.1380	0.1413	0.1412	0.1452	0.1444	0.1396	0.1364	0.1400	0.1424
	0.1408	0.1419	0.1415	0.1393	0.1472	0.1452	0.1387	0.1383	0.1267	0.1326
	0.1326	0.1398	0.1283	0.1291	0.1296	0.1282	0.1314	0.1235	0.1283	0.1179
	0.1206	0.1285	0.1365	0.1290	0.1290	0.1345	0.1191	0.1275	0.1290	0.1187
Z33	0.1252	0.1210	0.1268	0.1339	0.1333	0.1359	0.1309	0.1362	0.1315	0.1399
	0.1387	0.1369	0.1326	0.1381	0.1308	0.1301	0.1322	0.1302	0.1260	0.1241
	0.1266	0.1210	0.1298	0.1264	0.1232	0.1250	0.1313	0.1284	0.1257	0.1281
	0.1321	0.1350	0.1665	0.1695	0.1692	0.1386	0.1352	0.1422	0.1409	0.1332
Z34	0.1387	0.1343	0.1349	0.1335	0.1289	0.1300	0.1282	0.1263	0.1258	0.1331
	0.1268	0.1291	0.1353	0.1295	0.1304	0.1279	0.1345	0.1329	0.1329	0.1294
	0.1398	0.1386	0.1318	0.1278	0.1371	0.1317	0.1357	0.1361	0.1370	0.1416
	0.1291	0.1350	0.1368	0.1535	0.1340	0.1304	0.1312	0.1331	0.1276	0.1302
Z35	0.1232	0.1340	0.1316	0.1299	0.1375	0.1238	0.1344	0.1229	0.1331	0.1324
	0.1297	0.1297	0.1233	0.1286	0.1314	0.1334	0.1259	0.1362	0.1151	0.1279
	0.1256	0.1287	0.1323	0.1216	0.1263	0.1296	0.1241	0.1274	0.1252	0.1310
	0.1276	0.1314	0.1328	0.1284	0.1284	0.1339	0.1346	0.1360	0.1356	0.1359
Z41	9.7920	9.8090	9.8090	9.8130	9.8190	9.8730	9.7850	9.8220	9.7880	9.7530
	9.8170	9.7530	9.7060	9.7480	9.7840	9.7210	9.7330	9.7910	9.9090	9.9510
	9.9670	9.9340	9.8760	9.9070	9.9470	9.8780	9.9150	9.9200	9.9090	9.9220
	9.8440	9.8740	9.8000	9.8700	9.8970	9.8670	9.8760	9.8830	9.9370	9.9330
Z42	9.9070	9.8530	9.8510	9.8690	9.8250	9.8630	9.8610	9.8440	9.8500	9.7980
	9.8300	9.8250	9.8370	9.8890	9.8350	9.8030	9.7550	9.7960	9.7760	9.7730
	9.7270	9.6260	9.6430	9.6620	9.6920	9.6800	9.6990	9.3850	9.7020	9.7160
	9.7420	9.6530	9.7390	9.7830	9.7030	9.7460	9.7360	9.8000	9.7490	9.7840
Z43	9.7060	9.7540	9.7830	9.7500	9.7290	9.7900	9.7790	9.7370	9.7640	9.6970
	9.6850	9.7260	9.6830	9.6880	9.7230	9.7360	9.6930	9.7560	9.7500	9.7880
	9.7050	9.7660	9.7710	9.8240	9.8610	9.8290	9.8020	9.8550	9.7600	9.8230
	9.8610	9.8200	9.8420	9.8370	9.8340	9.8750	9.9040	9.8570	9.8000	9.8650
Z44	9.8190	9.8400	9.8350	9.7756	9.8520	9.8900	9.9230	9.8810	9.9580	9.9290
	9.9320	9.6500	9.9680	9.9220	9.8580	9.9460	9.8760	9.9400	9.8370	9.7400
	9.8990	9.9440	9.9570	10.0360	9.8960	9.9550	10.0230	10.0170	9.9950	9.7420
	9.6220	9.7320	9.7280	9.9780	10.1120	10.0350	9.9930	9.6710	9.5720	9.6780
Z45	9.7530	9.7570	9.7510	9.8330	9.7730	9.7980	9.8460	9.8440	9.8750	9.8690
	9.8300	9.6950	9.6930	9.6990	9.6540	9.6880	9.5790	9.6610	9.9250	9.8580
	9.6240	9.6830	9.8540	9.6300	9.5890	9.6450	9.7990	9.8260	9.9420	9.9150
	9.9150	9.7980	9.9240	9.8970	9.8820	9.8090	9.7990	9.8150	9.8580	9.8380

References

- Hall, D.L.; Llinas, J. An introduction to multisensor data fusion. *IEEE Proc.* **1997**, *85*, 6–23.
- Deng, X.; Jiang, W.; Zhang, J. Zero-sum matrix game with payoffs of Dempster–Shafer belief structures and its applications on sensors. *Sensors* **2017**, *17*, 922.
- Jin, X.; Du, J.; Bao, J. Maneuvering target tracking by adaptive statistics model. *J. China Univ. Posts Telecommun.* **2013**, *20*, 108–114.
- Sun, S.; Tian, T.; Lin, H. Optimal Linear Estimators for Systems with Finite-Step Correlated Noises and Packet Dropout Compensations. *IEEE Trans. Signal Process.* **2016**, *64*, 5672–5681.
- Jiang, W.; Wei, B.; Zhan, J.; Xie, C.; Zhou, D. A visibility graph power averaging aggregation operator: A methodology based on network analysis. *Comput. Ind. Eng.* **2016**, *101*, 260–268.
- Sun, S.; Wang, G. Modeling and estimation for networked systems with multiple random transmission delays and packet losses. *Syst. Control Lett.* **2014**, *73*, 6–16.
- Zhang, X.; Deng, Y.; Chan, F.T.S.; Adamatzky, A.; Mahadevan, S. Supplier selection based on evidence theory and analytic network process. *Proc. Inst. Mech. Eng Part B J. Eng. Manuf.* **2016**, *230*, 562–573.

8. Yang, F.; Ji, L.E.; Liu, S.; Feng, P. A fast and high accuracy registration method for multi-source images. *Optik* **2015**, *126*, 3061–3065.
9. Wu, F.; Wei, H.; Wang, X. Correction of image radial distortion based on division model. *Opt. Eng.* **2017**, *56*, 013108.
10. Yang, F.; Wei, H. Fusion of infrared polarization and intensity images using support value transform and fuzzy combination rules. *Infrared Phys. Technol.* **2013**, *60*, 235–243.
11. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353.
12. Jiang, W.; Xie, C.; Luo, Y.; Tang, Y. Ranking Z-numbers with an improved ranking method for generalized fuzzy numbers. *J. Intell. Fuzzy Syst.* **2017**, *32*, 1931–1943.
13. Bai, Y.; Zhang, B.; Wang, X.; Jin, X.; Xu, J.; Su, T.; Wang, Z. A novel group decision-making method based on sensor data and fuzzy information. *Sensors* **2016**, *16*, 1799.
14. Tang, Y.; Zhou, D.; Jiang, W. A new fuzzy-evidential controller for stabilization of the planar inverted pendulum system. *PLoS ONE* **2016**, *11*, e0160416.
15. Dempster, A.P. Upper and Lower Probabilities Induced by a Multi-valued Mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339.
16. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976; Volume 24.
17. Zadeh, L. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.* **1978**, *1*, 3–28.
18. Dubois, D.; Prade, H. *Possibility Theory: An Approach to Computerized Processing of Uncertainty*; Softcover Reprint of the Original, 1st ed; Plenum Press: New York, NY, USA, 1988.
19. Dubois, D.; Prade, H. Fuzzy sets and probability: Misunderstandings, bridges and gaps. In Proceedings of the 2nd IEEE International Conference on Fuzzy Systems, San Francisco, CA, USA, 28 March–1 April 1993; Volume 2, pp. 1059–1068.
20. Klir, G.J.; Yuan, B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*; Prentice Hall: Upper Saddle River, NJ, USA, 1995.
21. Dubois, D.; Prade, H. Possibility theory in information fusion. In Proceedings of the Third International Conference on Information Fusion, Paris, France, 10–13 July 2000; Volume 1, pp. PS6–PS19.
22. Jiang, W.; Zhan, J. A modified combination rule in generalized evidence theory. *Appl. Intell.* **2017**, *46*, 630–640.
23. Song, M.; Jiang, W.; Xie, C.; Zhou, D. A new interval numbers power average operator in multiple attribute decision making. *Int. J. Intell. Syst.* **2017**, *32*, 631–644.
24. Zadeh, L.A. A simple view of the Dempster–Shafer theory of evidence and its implication for the rule of combination. *AI Mag.* **1986**, *7*, 85–90.
25. Wang, J.; Xiao, F.; Deng, X.; Fei, L.; Deng, Y. Weighted evidence combination based on distance of evidence and entropy function. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 3218784.
26. Jiang, W.; Wang, S.; Liu, X.; Zheng, H.; Wei, B. Evidence conflict measure based on OWA operator in open world. *PLoS ONE* **2017**, *12*, e0177828.
27. Mo, H.; Lu, X.; Deng, Y. A generalized evidence distance. *J. Syst. Eng. Electron.* **2016**, *27*, 470–476.
28. Martin, A.; Jousselme, A.L.; Osswald, C. Conflict measure for the discounting operation on belief functions. In Proceedings of the 11th International Conference on Information Fusion, Cologne, Germany, 30 June–3 July 2008; pp. 1–8.
29. Smarandache, F.; Han, D.; Martin, A. Comparative study of contradiction measures in the theory of belief functions. In Proceedings of the 15th International Conference on Information Fusion, Singapore, 9–12 July 2012; pp. 271–277.
30. Martin, A. About conflict in the theory of belief functions. In *Belief Functions: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 161–168.
31. Deng, X.; Han, D.; Dezert, J.; Deng, Y.; Shyr, Y. Evidence combination from an evolutionary game theory perspective. *IEEE Trans. Cybern.* **2016**, *46*, 2070–2082.
32. Denoeux, T. A neural network classifier based on Dempster–Shafer theory. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2000**, *30*, 131–150.
33. Xiaobin, X.; Haishan, F.; Zhi, W.; Chenglin, W. An information fusion method of fault diagnosis based on interval basic probability assignment. *Chin. J. Electron.* **2011**, *20*, 255–260.
34. Tabassian, M.; Ghaderi, R.; Ebrahimpour, R. Combination of multiple diverse classifiers using belief functions for handling data with imperfect labels. *Expert Syst. Appl.* **2012**, *39*, 1698–1707.

35. Baudrit, C.; Dubois, D. Practical representations of incomplete probabilistic knowledge. *Comput. Stat. Data Anal.* **2006**, *51*, 86–108.
36. Mönks, U.; Dörksen, H.; Lohweg, V.; Hübner, M. Information fusion of conflicting input data. *Sensors* **2016**, *16*, 1798.
37. Mönks, U. *Information Fusion under Consideration of Conflicting Input Signals*; Springer: Berlin/Heidelberg, Germany, 2017.
38. Zhang, X.; Mahadevan, S.; Deng, X. Reliability analysis with linguistic data: An evidential network approach. *Reliab. Eng. Syst. Saf.* **2017**, *162*, 111–121.
39. Jiang, W.; Xie, C.; Zhuang, M.; Shou, Y.; Tang, Y. Sensor Data Fusion with Z-Numbers and Its Application in Fault Diagnosis. *Sensors* **2016**, *16*, 1509.
40. Guo, H.; Shi, W.; Deng, Y. Evaluating sensor reliability in classification problems based on evidence theory. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2006**, *36*, 970–981.
41. Yuan, K.; Xiao, F.; Fei, L.; Kang, B.; Deng, Y. Modeling sensor reliability in fault diagnosis based on evidence theory. *Sensors* **2016**, *16*, 113.
42. Glock, S.; Voth, K.; Schaede, J.; Lohweg, V. A framework for possibilistic multi-source data fusion with monitoring of sensor reliability. In Proceedings of the World Conference on Soft Computing, San Francisco, CA, USA, 23–26 May 2011.
43. Ehlenbröker, J.F.; Mönks, U.; Lohweg, V. Sensor defect detection in multisensor information fusion. *J. Sens. Sens. Syst.* **2016**, *5*, 337–353.
44. Zadeh, L.A. A Note on Z-numbers. *Inf. Sci.* **2011**, *181*, 2923–2932.
45. Deng, X.; Xiao, F.; Deng, Y. An improved distance-based total uncertainty measure in belief function theory. *Appl. Intell.* **2017**, *46*, 898–915.
46. Fu, C.; Yang, J.; Yang, S. A group evidential reasoning approach based on expert reliability. *Eur. J. Oper. Res.* **2015**, *246*, 886–893.
47. Jiang, W.; Xie, C.; Zhuang, M.; Tang, Y. Failure mode and effects analysis based on a novel fuzzy evidential method. *Appl. Soft Comput.* **2017**, *57*, 672–683.
48. Smets, P.; Kennes, R. The transferable belief model. *Artif. Intell.* **1994**, *66*, 191–234.
49. Jiang, W.; Zhuang, M.; Xie, C.; Wu, J. Sensing attribute weights: A novel basic belief assignment method. *Sensors* **2017**, *17*, 721.
50. Casella, G.; Berger, R. *Statistical Inference*; Duxbury Press: Pacific Grove, CA, USA, 2001.
51. Wu, X.; Kumar, V.; Quinlan, J.R.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Liu, B.; Yu, P.S.; et al. Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37.
52. Wen, C.; Xu, X. *Theories and Applications in Multi-Source Uncertain Information Fusion—Fault Diagnosis and Reliability Evaluation*; Beijing Science Press: Beijing, China, 2012.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Online Denoising Based on the Second-Order Adaptive Statistics Model

Sheng-Lun Yi ^{1,2}, Xue-Bo Jin ^{1,2}, Ting-Li Su ^{1,2,*}, Zhen-Yun Tang ³, Fa-Fa Wang ^{1,2}, Na Xiang ^{1,2} and Jian-Lei Kong ^{1,2}

- ¹ School of Computer Information and Engineering, Beijing Technology and Business University, Beijing 100048, China; yishenglun@st.btbu.edu.cn (S.-L.Y.); jinxuebo@btbu.edu.cn (X.-B.J.); wangfafa@st.btbu.edu.cn (F.-F.W.); xiangna@st.btbu.edu.cn (N.X.); kongjianlei@btbu.edu.cn (J.-L.K.)
 - ² Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology and Business University, Beijing 100048, China
 - ³ The Key Laboratory of Urban Security and Disaster Engineering, Ministry of Education, Beijing University of Technology, Beijing 100124, China; tzy@bjut.edu.cn
- * Correspondence: sutingli@btbu.edu.cn; Tel.: +86-10-6898-5241

Received: 24 April 2017; Accepted: 10 July 2017; Published: 20 July 2017

Abstract: Online denoising is motivated by real-time applications in the industrial process, where the data must be utilizable soon after it is collected. Since the noise in practical process is usually colored, it is quite a challenge for denoising techniques. In this paper, a novel online denoising method was proposed to achieve the processing of the practical measurement data with colored noise, and the characteristics of the colored noise were considered in the dynamic model via an adaptive parameter. The proposed method consists of two parts within a closed loop: the first one is to estimate the system state based on the second-order adaptive statistics model and the other is to update the adaptive parameter in the model using the Yule–Walker algorithm. Specifically, the state estimation process was implemented via the Kalman filter in a recursive way, and the online purpose was therefore attained. Experimental data in a reinforced concrete structure test was used to verify the effectiveness of the proposed method. Results show the proposed method not only dealt with the signals with colored noise, but also achieved a tradeoff between efficiency and accuracy.

Keywords: online denoising; the second-order adaptive statistics model; Kalman filter; Yule–Walker algorithm; real-time data processing

1. Introduction

With recent improvements in sensor technologies, information networks, and telemetry, an enormous amount of data is collected every day. At the same time, with the help of data processing techniques, policy-makers and scientists are now able to deploy these sampled data in significant applications such as target location [1], disease case count prediction [2], structural health monitoring [3], and financial forecasting [4]. However, signals may be subject to random noise in practical processes, due to such reasons as incorrect measurements, faulty sensors, or imperfect data collection. Any noise and instability can be considered as the source of error, which would result in signal distortion.

How to eliminate the influence of the noise in measured data and extract the useful information has been a focus of information science research. Currently, existing algorithms primarily focus on the offline denoising problem, which requires a full set of data to accomplish the denoising process. Common solutions can be divided into two categories, i.e., offline denoising in the time domain [5–7] and in the frequency domain [8]. Specifically, in the time domain, Weissman et al. [5] proposed the discrete universal denoiser (DUDE) algorithm for offline denoising. DUDE assumes the statistical

knowledge of the noise mechanism, but makes no assumptions on the distribution of the underlying data. Furthermore, Motta et al. [6] presented an extension of the DUDE specialized for the denoising of grayscale images. Moon et al. [7] introduced S-DUDE, a new algorithm for denoising discrete memoryless channel(DMC)-corrupted data. Aside from DUDE, the wavelet transform method is another classical offline method. Various offline denoising techniques based on the modified wavelet transform method can be found in different denoising-related applications [9–11]. The methods above possess strong capabilities for removing noise and preserving data details. However, the commonality is that they require all the data as a necessary condition.

In the frequency domain, Bhati et al. [12] designed a time–frequency localized three-band biorthogonal linear phase wavelet filter bank for epileptic seizure electroencephalograph (EEG) signal classification. For nonlinear and nonstationary signals, Gao et al. [13] proposed a novel framework of amalgamating empirical mode decomposition (EMD) with variable regularized two-dimensional sparse non-negative matrix factorization (v-SNMF2D) for single-channel source separation. The method can avoid certain strong constraints in separating blind source signals, and provide a robust sparse decomposition. However, it also needs all the data for the denoising process, and is not suitable for online purposes. Yin et al. [8] presented a novel approach for mechanical vibration signal denoising filter using partial differential equation(PDE) and its numerical solution. This method is not only easy to achieve but also can obtain smooth filtering results. However, this method has some limitations. First, the cut-off frequency in practical systems is difficult to determine. Second, as the amount of data increases, the arithmetic speed is slower, and the inverse matrix will be out of memory.

In real-world applications, such as condition monitoring [14], critical event forecasting [2] or health monitoring [3], data is susceptible to the noise and instability of the measurement process. Existence of serious noise in real-time data may cause not only the inaccurate outcomes but also the failure of the entire system. Meanwhile, in many physical systems, we need to collect and process data in real time. Therefore, online denoising is very necessary in data processing.

On one hand, smoothing and trend filtering methods are typically used in data processing to remove useless information. The moving average method is a well-known and popular technique due to its simplicity for online denoising [15] and long-term forecasting applications [16,17]. However, when there is too much random impulse disturbance in the experiment process, the measured data will show a strong maneuvering feature, and the smoothing filtering is not applicable to deal with it. In recent years, further study has improved these canonical methods. Exponential smoothing was used in prediction applications such as tourism demand [18], composite index data [19] or inflation rate [20]. Jere et al. [20] described the Holt’s exponential smoothing algorithm based on the assumption of a model consisting of a trend. Recent observations are expected to have significant influence on the future values in a series. Therefore, the prediction of the future value could be implemented through some previous points. Online denoising could also be achieved in a similar way. For instance, Goh et al. [21] proposed a sequential myriad smoothing approach for tracking a time-varying location parameter corrupted by impulsive symmetric α stable noise.

On the other hand, the Kalman filter, as a popular estimation method, has been widely used in various online applications, like navigation [22], target tracking [23], and signal processing [24], etc. The most important advantage is that it can not only retain the useful information but also obtain the most optimal estimate online. For example, the Kalman filter is used to minimize the error in stereo vision-based distance measurement data (3D position of pedestrians) in [25]. Huang et al. [26] developed the noise reduction method by a hybrid Kalman filter with an autoregressive moving average (ARMA) model. The coefficients of the AR model for the Kalman filter are calculated by solving for the minimum square error solutions. Rosa et al. [27] presented a Kalman filter-based approach for track reconstruction in a neutrino telescope, which can effectively remove the errors caused by noise and improve the accuracy of the data.

It needs to be pointed out that, when using the Kalman filter, an accurate system dynamic model would offer great help to achieve the optimal estimation. Miao et al. [28] used the Kalman filter with several different kinds of system models to remove the noise of the storage volume data of the internet center. Due to the difficulty in obtaining the density characteristic of the practical data, the adaptive model was proposed to capture the characteristics of the moving targets in [29], and estimate the acceleration based on the adaptive parameter.

In this paper, a denoising method for real-time data with unstable fluctuation and colored noise was investigated. For the sake of the data features and the online requirement, the Kalman filtering method based on a second-order adaptive statistics model was proposed here, and its performance was verified by some real test data. Moreover, the test data was processed via another two representative methods: first-order exponential smoothing [18] and Holt's exponential smoothing [20], and the results demonstrated that the proposed method could give a better effect.

Compared to previous works, the contribution of this work is that we used a second-order adaptive model for online denoising, which can obtain a better denoising performance for the measurements in the reinforced concrete structure test experiment. The comparison between our model and the third-order model [29] is given in Section 3, and the results show that the developed second-order adaptive model here can obtain a smaller error and consume less time.

The structure of this paper is as follows. Section 2 presents the specific method of the second-order adaptive statistics model. The overview of the experiment is provided in Section 3. Section 4 discusses the robustness and the real-time performance. Some conclusions are given in Section 5.

2. Online Denoising Algorithm Based on Kalman Filtering and the Adaptive Statistics Model

For the purpose of removing the unexpected noise in an online mode, Kalman filtering was actually a competitive solution, where only the estimation derived in the previous step and the measurements in the current step were required to compute the new estimated values. However, this is not enough to obtain the desired results. A reasonable model that could describe the dynamic features of the data is another impact factor in the denoising process. Therefore, a second-order adaptive statistics model is presented later in this section, and the method to compute the adaptive parameter is explained in detail as well.

2.1. Online Denoising Algorithm Based on Kalman Filtering

Kalman filtering is one of the most classical recursive algorithms that gives the optimal estimation of the state vector. The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: state update equation and measurement update equation, which can be expressed as:

$$\begin{aligned}x(k+1) &= \Phi(k+1|k)x(k) + U(k)u(k) + w(k) \\z(k+1) &= H(k+1)x(k+1) + v(k+1)\end{aligned}\quad (1)$$

where x is the state vector of the system to be estimated, whose initial value and covariance are known as x_0 and P_0 . $\Phi(k+1|k)$ is the state-transition matrix. $u(k)$ is the system input and $U(k)$ is the corresponding matrix. $w(k)$ and $v(k)$ are the process noise and measurement noise respectively, and the variance of $v(k)$ is known (as R). Note that both $w(k)$ and $v(k)$ are white noise with zero mean and independent of the initial state x_0 . $z(k)$ is the measurement vector and $H(k)$ is the observation matrix.

The Kalman filtering considers the correlation between errors in the prediction and the measurements. The algorithm is in a predict-correct form, which is convenient for implementation as follows:

(1) Initialization:

$$\hat{x}(0|0) = x_0 \quad P(0|0) = P_0 \quad (2)$$

(2) Prediction:

$$\hat{x}(k+1|k) = \Phi(k+1|k)\hat{x}(k|k) + U(k)u(k) \quad (3)$$

$$P(k+1|k) = \Phi(k+1|k)P(k|k)\Phi^T(k+1,k) + Q(k) \quad (4)$$

(3) Correction:

$$K(k+1) = P(k+1|k)H^T(k+1)[H(k+1)P(k+1|k)H^T(k+1) + R]^T \quad (5)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)[y(k+1) - H(k+1)\hat{x}(k+1|k)] \quad (6)$$

$$P(k+1|k+1) = [I - K(k+1)H(k+1)]P(k+1|k) \quad (7)$$

According to the equations above. The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. Since the algorithm can run recursively, we can implement it step by step, that is, the denoised data can be obtained in real time.

2.2. Adaptive Statistics Model for Online Denoising

Considering the unstable fluctuation of the data and the existence of colored noise, the linear time-invariant model with noise as used in Section 2.1 may not be suitable for describing this kind of data. Therefore, we proposed a second-order adaptive statistics model to deal with these challenges. Let x , \dot{x} be the data itself and the gradient, respectively. The state vector is expressed as $x = [x, \dot{x}]^T$ throughout this paper unless stated otherwise explicitly.

Referring to the colored noise, it mainly lies in the changing process of the data gradient. When the data is varying with time, its gradient will follow certain rule: value of the gradient at the next time tick is always within the neighborhood of the current predicted gradient value. Therefore, the gradient can be computed as:

$$\dot{x}(t) = \bar{g}(t) + \Delta(t) \quad (8)$$

where $\bar{g}(t)$ is the predicted value of $\dot{x}(t)$ in current interval. In particular, $\Delta(t)$ stands for the maneuvering change with colored noise. Considering that Kalman filter has specific requirements for the type of the noise, colored noise in $\Delta(t)$ needs to be processed. Therefore, the Wiener-Khinchin theorem was introduced here, which assumes it corresponds to the first-order stationary Markov process:

$$\dot{\Delta}(t) = -\alpha\Delta(t) + w(t) \quad (9)$$

where α is the parameter of maneuvering frequency [29], and $w(t)$ is a Gaussian white noise with zero mean and a variance of σ_w^2 . With the two equations above, the change of the gradient can be written as:

$$\ddot{x}(t) = -\alpha\dot{x}(t) + \alpha\bar{g}(t) + w(t) \quad (10)$$

since $\ddot{x}(t) = \dot{\Delta}(t)$ over any sampling interval.

Therefore, the state-space representation of the continuous-time adaptive model is:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -\alpha \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \alpha \end{bmatrix} \bar{g}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t) \quad (11)$$

Let $A = \begin{bmatrix} 0 & 1 \\ 0 & -\alpha \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ -\alpha \end{bmatrix}$, $C = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. The solution of equations is:

$$x(t) = e^{At}x(t_0) + \int_0^t e^{A(t-\lambda)}B\bar{g}(\lambda)d\lambda + \int_0^t e^{A(t-\lambda)}Cw(\lambda)d\lambda \tag{12}$$

We assume $t = t_0 + T$ and $t_0 = kT$. Then we can get the discrete-time equivalent as the following:

$$x(k + 1) = \Phi(k + 1|k)x(k) + U(k)\bar{g}(k) + w(k) \tag{13}$$

With Laplace transforms, matrix $\Phi(k + 1|k)$ can be expressed as:

$$\Phi(k + 1|k) = e^{AT} = \begin{bmatrix} 1 & \frac{(1-e^{-\alpha T})}{\alpha} \\ 0 & e^{-\alpha T} \end{bmatrix} \tag{14}$$

Matrix $U(k)$ can be described as:

$$U(k) = \int_0^T e^{A(T-\lambda)}Bd\lambda = \begin{bmatrix} T \\ 1 \end{bmatrix} - \begin{bmatrix} \frac{(1-e^{-\alpha T})}{\alpha} \\ e^{-\alpha T} \end{bmatrix} = \begin{bmatrix} T - \frac{1-e^{-\alpha T}}{\alpha} \\ 1 - e^{-\alpha T} \end{bmatrix}. \tag{15}$$

The variance of the $w(k)$ can be computed in the following way:

$$Q(k) = E[w(k)w^T(k)] = \int_0^T e^{A(T-\lambda)}C\sigma_\Delta^2Ce^{A(T-\lambda)}d\lambda = 2\alpha\sigma_\Delta^2 \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \tag{16}$$

where

$$\begin{aligned} q_{11} &= \frac{1}{2\alpha^3} (4e^{-\alpha T} - 3 - e^{-2\alpha T} + 2\alpha T) \\ q_{12} &= \frac{1}{2\alpha^2} (e^{-2\alpha T} + 1 - 2\alpha T) \\ q_{22} &= \frac{1}{2\alpha} (1 - e^{-2\alpha T}) \end{aligned} \tag{17}$$

2.3. Adaptive Parameter Adjustment via the Yule-Walker Algorithm

In the previous subsection, a statistics model was presented to capture the fluctuation features in the measured data. It needs to be pointed out that in the proposed model, the adaptive parameter α is not only unknown, but also self-adaptive.

We adopted the following method to update parameter α and σ_Δ^2 based on the Yule-Walker estimated algorithm [29]. First of all, we need to discretize the Equation (9). Through substituting A to $-\alpha$ and C to 1 in Equations (14) and (16), we can obtain its discrete-time equivalent:

$$\Delta(k) = \beta\Delta(k - 1) + w(k - 1) \tag{18}$$

where $w(k - 1)$ is a discrete-time zero-mean white noise sequence with variance $\sigma_{\Delta\omega}^2 = \sigma_\Delta^2(1 - \beta^2)$ and $\beta = e^{-\alpha T}$. Then, the method of parametric update is as follows:

(1) Initialization:

$$\alpha(0) = \alpha_0 \quad \sigma_\Delta^2(0) = \sigma_{\Delta 0}^2 \quad r_0(0) = \dot{x}_0 \cdot \dot{x}_0 \quad r_0(1) = \dot{x}_0 \tag{19}$$

(2) Set the estimation of gradient and $\bar{g}(k)$ as:

$$\hat{x}(k) = \dot{x}(k|k) \tag{20}$$

$$\bar{g}(k) = \frac{1}{k} \sum_{k=0}^k \hat{x}(k|k) \tag{21}$$

The parameter of $\sigma(k)$ is satisfied with the first-order stationary Markov process:

$$\beta(k) = \frac{r_k(1)}{r_k(0)} \quad \sigma_{\Delta\omega}^2(k) = r_k(0) - \alpha(k)r_k(1) \tag{22}$$

(3) Parameter update:

$$\begin{aligned} r_k(1) &= r_{k-1}(1) + \frac{1}{k} [\hat{x}(k)\hat{x}(k-1) - r_{k-1}(1)] \\ r_k(0) &= r_{k-1}(0) + \frac{1}{k} [\hat{x}(k)\hat{x}(k) - r_{k-1}(0)] \end{aligned} \tag{23}$$

and

$$\begin{aligned} \alpha(k) &= -\frac{\ln r_k(1) - \ln r_k(0)}{T} \\ \sigma_{\Delta}^2(k) &= \frac{r_k(0) - \alpha(k)r_k(1)}{1 - \left(\frac{r_k(1)}{r_k(0)}\right)^2} \end{aligned} \tag{24}$$

Then, we can use the Equation (24) to get α and σ_{Δ}^2 so that we can achieve the purpose of updating the system parameters.

Using the method described in this section, online denoising of data with unstable fluctuation and colored noise was then accomplished. The flow chart of the proposed method was given in Figure 1. It can be seen that the method consists of two parts within a closed loop. The first one is to estimate the system state with the Kalman filter based on the second-order adaptive statistics model, and the other is to update the adaptive parameter in the model by the Yule–Walker algorithm. In the next section, the effectiveness of this method will be evaluated via the experiment data from a reinforced concrete structure test, and the results will also be compared to some other representative online denoising methods.

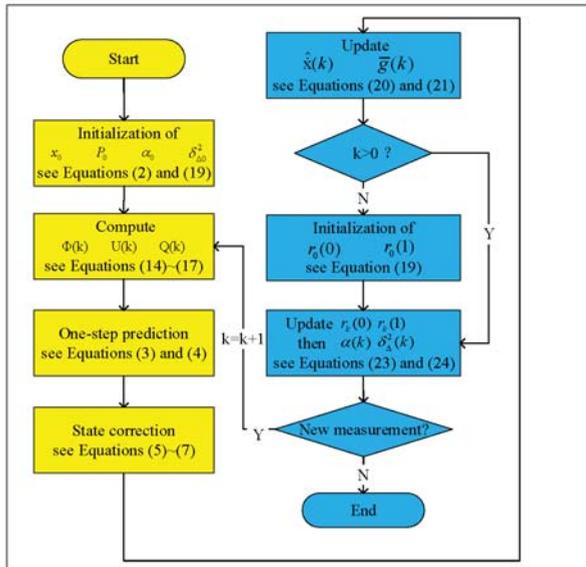


Figure 1. The flow chart of the proposed online denoising method.

3. Experiments

In order to verify the effectiveness of the proposed algorithm, experimental data from the test of a reinforced concrete structure was adopted. The configuration of the experiment is shown in Figure 2. It was a quasi-static test for the column made by Chinese Grade 345 steel and C30 Grade

concrete [30]. During the experiment, the column was tested under constant axial load and cyclic bending. Through this experiment, deformation displacement at different time samples were obtained, which correspond to the measurements in the proposed algorithm. Although the entire data was ready before denoising as well, the process was implemented in an ‘online’ mode, i.e., only the measurement of the ‘current’ sampling time and previous result would be used in computation. The necessity of the online mode for this background is because the actual value of the measured state has great effect on the identification of the structure security, and it needs to be known during the monitoring process. In this experiment, the sampling time was set as 0.001 s.

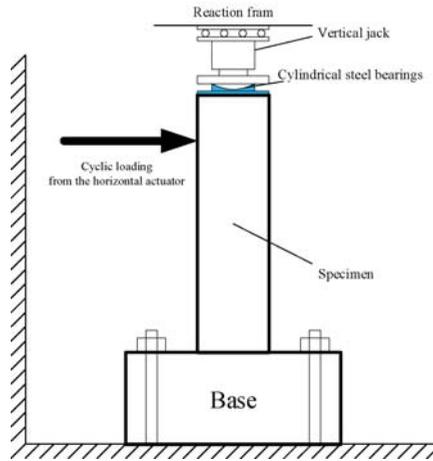


Figure 2. The configuration of the experiment.

Figure 3 gives the measurement and the real data, which is used to test the performance of the developed method. The measurement data came from the experiment and the real data came from the offline filter with high degree of accuracy. As can be clearly seen from Figure 3, the measured data possessed a unstable fluctuation as well as the existence of the colored noise.

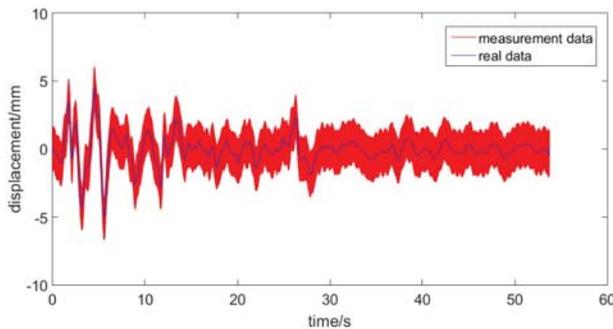


Figure 3. The real data and measurement data.

In this paper, we compared a second-order adaptive statistics model with various other methods such as first-order exponential filtering, Holt’s exponential filtering or a third-order adaptive statistics model to deal with the denoising problem for the real-time deformation displacement data. In order to evaluate these methods, mean and covariance of the error were compared. In addition, the

root-mean-square error (RMSE) was used. The RMSE is very commonly used and makes for an excellent general purpose error metric for numerical predictions. Specifically, 'mean' here represents averaged absolute value of difference between the real data and the denoised data, i.e.,

$$\text{mean} = \frac{\sum_{i=1}^n |r_i - d_i|}{n} \quad (25)$$

where n is the number of the measurements, r_i is the i th real data and d_i is the corresponding denoised data.

Then, the covariance is defined as the following:

$$\text{cov} = \frac{\sum_{i=1}^n (\text{mean} - |r_i - d_i|)^2}{n} \quad (26)$$

Finally, the RMSE can be expressed as the following:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (d_i - r_i)^2}{n}} \quad (27)$$

In the following context, three cases are implemented. In the first two cases the comparison between different denoising methods is depicted, while a third case is given to discuss the effect of the initial value on the denoising performance. In Section 3.1, the adaptive statistics models, including the second-order model and the third-order model, were used to deal with the data; in Section 3.2, we compared the developed method with the first-order exponential filtering and Holt's exponential filtering, respectively; in Section 3.3, through eliminating data within the adjustment process and retaining the posterior convergent data, the denoising effect was obviously improved.

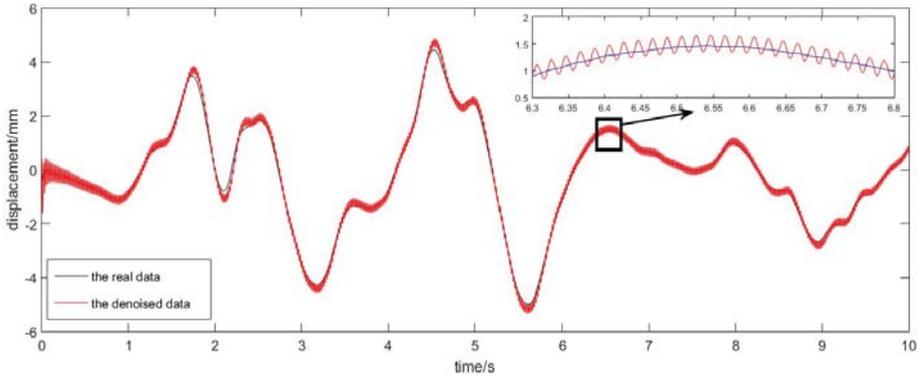
3.1. The Denoising Effect of the Adaptive Statistics Model

The performances of the second-order and the third-order adaptive methods for online denoising were compared in this part. The denoised results are shown in Figure 4. Since the difference was too small, we provide a detailed part of the curves in the small picture, and 500 points from 6.3 s to 6.8 s are shown there. The results demonstrated that this algorithm is feasible and reliable with reasonable precision. Furthermore, through comparing the real data and the denoised data, the satisfactory denoising effect of the second-order adaptive statistics model was illustrated.

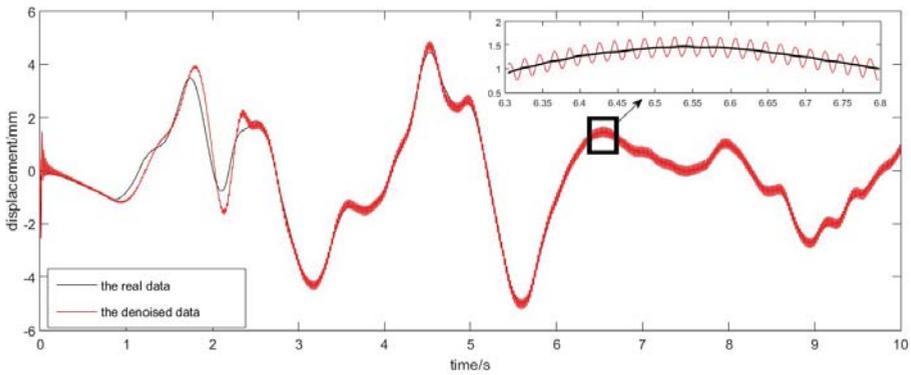
Comparing the second-order and third-order adaptive statistics models, we can find a satisfactory denoising effect in Figure 4a,b. However, from the result before 3 s, we might notice the third-order adaptive statistics model performs with poorer precision. Thus, the second-order adaptive statistics model can have advantages with respect to accuracy. Meanwhile, in order to better describe the error and compare the denoising precision, Figure 5 gives the error of the both models.

The results in Figure 5 show that the second-order adaptive statistics model has the smaller error. In order to better prove this conclusion, more groups of data were adopted to test the method, and each group contained 10,000 points. The symbol $mean_m$ here represents the mean of the whole data set. The results of the tests are shown in Table 1. Obviously, for each group, results from the second-order model all showed better performance both in mean, covariance and RMSE. As a whole, variance and RMSE of the second-order model was only about 0.0223 and 0.1461, respectively, better than that of the third-order model (0.1407 and 0.3129). On the other hand, Kalman filtering is an estimation algorithm which shows resemblance and proximity with the one-step prediction. We can estimate next step value by merely using the last measurement. Therefore, it is an online algorithm, that is, there is the negligible delay with the denoising process. In addition, the calculated amount of the second-order model is lower than for the third-order model. This is due to the more computational expense caused

by the larger matrices in the higher-order model. Therefore, results showed the second-order adaptive statistics model could not only deal with the signals with colored noise in real time, but also achieve a tradeoff between efficiency and accuracy.

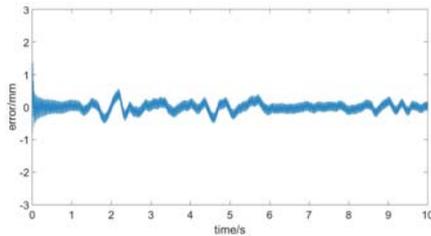


(a) The denoised result of the second-order adaptive statistics model.

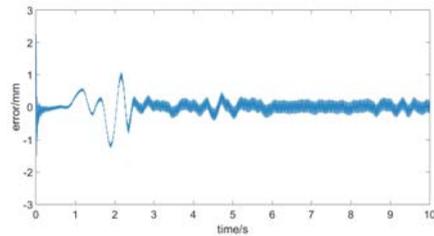


(b) The denoised result of the third-order adaptive statistics model.

Figure 4. The denoised result of the adaptive statistics models.



(a) Error for the second-order adaptive statistics model.



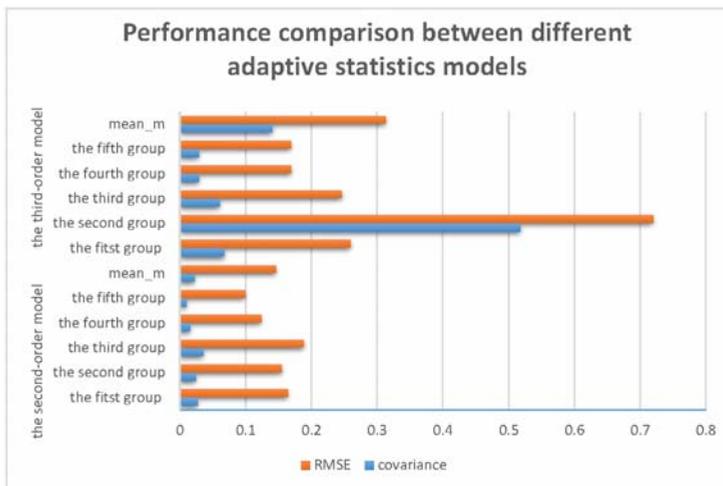
(b) Error for the third-order adaptive statistics model.

Figure 5. The error comparison of the adaptive statistics models for online denoising.

Table 1. Performance comparison between different adaptive statistics models.

	Second-Order Adaptive Statistics Model			Third-Order Adaptive Statistics Model		
	Mean/mm	Covariance	RMSE	Mean/mm	Covariance	RMSE
<i>First group</i>	0.1301	0.0269	0.1640	0.1762	0.0675	0.2598
<i>Second group</i>	0.1130	0.0241	0.1552	0.4852	0.5179	0.7197
<i>Third group</i>	0.1375	0.0355	0.1884	0.1757	0.0607	0.2463
<i>Fourth group</i>	0.0930	0.0153	0.1237	0.1107	0.0287	0.1694
<i>Fifth group</i>	0.0721	0.0098	0.0990	0.1179	0.0286	0.1691
<i>Mean_m</i>	0.1091	0.0223	0.1461	0.2131	0.1407	0.3129

Based on the results in the Table 1, it can be clearly seen that the second-order adaptive statistics model is better than the third-order one, because it provided better precision and faster speed in online denoising. Meanwhile, as we can see in Figure 6, more stable denoising effect and smaller RMSE can be offered by the second-order statistics model, in which the ‘orange column’ is the RMSE and the ‘blue column’ is the covariance for each group.

**Figure 6.** Covariance and RMSE of the adaptive statistics models.

3.2. Comparison of the Denoising Effect between the Proposed Method and the Exponential Smoothing

Formerly, the exponential smoothing was typically for forecasting. Simultaneously, it could also be applied in online denoising [21]. When using the exponential smoothing, parameter selection is very important, as it can adjust the development tendency of the data trend. However, it is usually very subjective. Nowadays, the primary methods for parameters selection can be divided into two ways: one is the empirical method, the other is trial method. In this paper, we adopted the empirical method. Finally, we decided to utilize first-order exponential smoothing and Holt’s exponential smoothing for comparison with the result in Section 3.1.

3.2.1. The Denoising Effect of the First-Order Exponential Smoothing

We utilized priori knowledge to select the parameters of 0.2, 0.5 and 0.8. A first-order exponential smoothing with different parameters was used to denoise the same five groups of data as those in Section 3.1, and the results are given in Table 2. According to those test results, we can draw a

conclusion that first-order exponential smoothing [18] with a parameter of 0.2 possessed the best denoising effect.

Table 2. Mean, covariance and RMSE of first-order exponential smoothing with different parameters.

Various Models	The Parameter of 0.2			The Parameter of 0.5			The Parameter of 0.8		
	Mean	Covariance	RMSE	Mean	Covariance	RMSE	Mean	Covariance	RMSE
First group	0.5864	0.4352	0.6597	0.9282	1.0813	1.0399	1.0112	1.2645	1.1245
Second group	0.5867	0.4351	0.6596	0.9284	1.0814	1.0340	1.0113	1.2642	1.1244
Third group	0.5870	0.4353	0.6598	0.9287	1.0813	1.0399	1.0118	1.2637	1.1241
Fourth group	0.5861	0.4340	0.6588	0.9279	1.0794	1.0389	1.0108	1.2617	1.1233
Fifth group	0.5864	0.4340	0.6588	0.9282	1.0802	1.0393	1.0112	1.2625	1.1236
Mean _m	0.5865	0.4347	0.6593	0.9283	1.0807	1.0384	1.0113	1.2633	1.1240

3.2.2. The Denoising Effect of the Holt's Exponential Smoothing

Within Holt's exponential smoothing [20], two kinds of states were usually used: one was the signal of the backward-smoothing, and the other was the tendency of the backward-smoothing. As a result, we introduced two parameters a and b . b was set to be 0.8 as empirical value, meanwhile, the parameter a was selected the same as the first-order exponential smoothing method, which was 0.2, 0.5 and 0.8. The same data was used as before, and the results are shown in Table 3.

Table 3. Mean, covariance and RMSE of the Holt's exponential smoothing with different parameter a .

Various Models	The Parameter a of 0.2			The Parameter a of 0.5			The Parameter a of 0.8		
	Mean	Covariance	RMSE	Mean	Covariance	RMSE	Mean	Covariance	RMSE
First group	0.3832	0.1891	0.4349	0.5978	0.4679	0.6840	0.9326	1.0542	1.0267
Second group	0.3824	0.1870	0.4324	0.5973	0.4669	0.6833	0.9329	1.0532	1.0262
Third group	0.3824	0.1870	0.4324	0.5975	0.4667	0.6832	0.9326	1.0532	1.0262
Fourth group	0.3816	0.1861	0.4314	0.5965	0.4655	0.6823	0.9321	1.0505	1.0249
Fifth group	0.3818	0.1857	0.4309	0.5968	0.4657	0.6824	0.9324	1.0512	1.0253
Mean _m	0.3823	0.1870	0.4324	0.5972	0.4665	0.6830	0.9325	1.0525	1.0259

It can be clearly seen in Table 3 that the best denoising effect can be acquired with the parameter a of 0.2 and b of 0.8, but the value of different indicators was still obviously larger than those of the proposed adaptive method. Table 4 gives a summary of performance comparison among different methods.

Table 4. The results by several kinds of online denoising methods.

Various models	Variou orders/parameters	Mean _m of mean/mm	Mean _m of covariance	Mean _m of RMSE
Adaptive statistics model	Second-order	0.1091	0.0223	0.1461
	Third-order	0.2131	0.1407	0.3129
First-order exponential smoothing	Parameter of 0.2	0.5865	0.4347	0.6593
	Parameter of 0.5	0.9283	1.0807	1.0384
	Parameter of 0.8	1.0113	1.2633	1.1240
Holt's exponential smoothing	Parameter a of 0.2	0.3823	0.1870	0.4324
	Parameter a of 0.5	0.5972	0.4665	0.6830
	Parameter a of 0.8	0.9325	1.0525	1.0259

In these three categories of online denoising methods, the mean, covariance and RMSE of the adaptive statistics model are obviously the smallest. The results indicated that online denoising could be better achieved via the adaptive statistics model, because the system parameter could be adjusted dynamically as the denoising process was implemented. Furthermore, by contrasting the second-order adaptive model and the third-order adaptive model, we have come to the tentative conclusion that the effect of the second-order adaptive model is more outstanding. To sum up, between the two

exponential smoothing methods, the Holt's exponential smoothing with the parameter a of 0.2 and b of 0.8 has better denoising effect. However, among all the different methods conducted in this paper, the second-order adaptive statistics model presented the best performance. It not only showed good denoising accuracy, but also gave a faster processing speed.

3.3. The Effect of Initial Value on the Denoising Performance

In this case, we would analyze the figure of the error data, as shown in Figure 7.

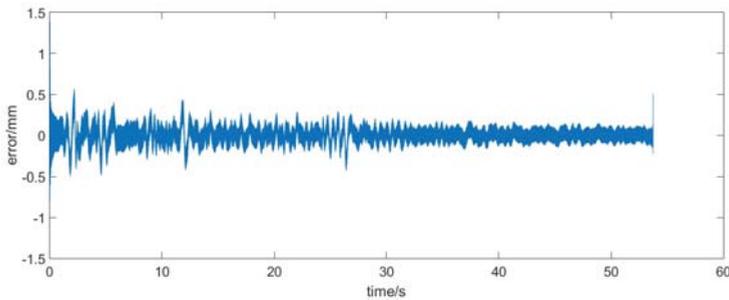


Figure 7. The real data and measurement data.

From the figure above it can be clearly seen that online denoising based on the adaptive statistics model had a regulatory process at the beginning. This is because the initial value of x_0 was zero and P_0 was very big. It thus appears that we could obtain the more precise filtering results through the index for selection. Actually, it needs to be emphasized that the convergence procedure existed in the adaptive model, that is, the denoising effect is better as time goes on. Finally, we selected the last 5000 points to calculate the covariance and the mean.

As can be clearly seen from the Table 5 and Figure 8, mean, covariance and RMSE decreased significantly compared with those in Table 1 and Figure 6. By assessing the data, the covariance of the second-order model is only 0.0171 and RMSE is only 0.1200, while for the third-order model these values are 0.0345 and 0.1760, respectively. Recall that the best filter effect of exponential smoothing is about 0.2 and 0.43. This leads one to believe that the adaptive statistics model was superior to the exponential smoothing. When comparing two approaches using the adaptive statistics models, we can find the denoising effect of the second-order adaptive statistics model is better than the third-order adaptive statistics model. This is because the general trend of data seems more consistent with the second order.

In fact, except for precision, the second-order model has another preponderance. It possesses a smaller computation burden. We computed the runtime for each denoising process, and found the second-order adaptive model is faster than the smoothing filter and the third-order model. If we started to denoise the data with 52,741 counts, the elapsed time of the second-order model is 9.142300 s. On the contrary, we need 13.124500 s for the third-order model. Considering the statements above, we can come to the conclusion that the second-order adaptive statistics model is kind of more accurate and efficient method to proceed online denoising.

Table 5. Mean, covariance and RMSE of the last 5000 points derived by the adaptive statistics model.

Various Model	Second-Order Adaptive Statistics Model			Third-Order Adaptive Statistics Model		
	Mean	Covariance	RMSE	Mean	Covariance	RMSE
<i>First group</i>	0.1106	0.0178	0.1334	0.1333	0.0235	0.1533
<i>Second group</i>	0.0888	0.0120	0.1095	0.2301	0.0808	0.2842
<i>Third group</i>	0.1528	0.0454	0.2131	0.1343	0.0304	0.1743
<i>Fourth group</i>	0.0682	0.0064	0.0800	0.1260	0.0269	0.1640
<i>Fifth group</i>	0.0536	0.0041	0.0640	0.0842	0.0109	0.1044
<i>Mean_m</i>	0.0948	0.0171	0.1200	0.1416	0.0345	0.1760

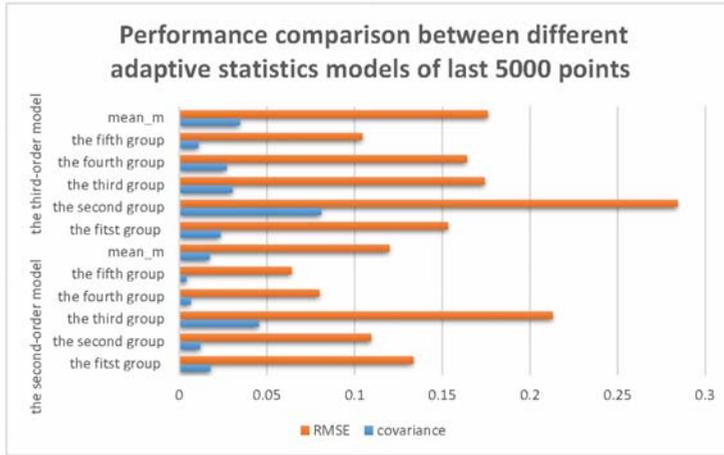


Figure 8. Covariance and RMSE of the adaptive statistics models of the last 5000 points.

4. Discussion

In the previous section, through the experiment data and the comparison with other classical denoising methods, the effectiveness and superiority of the proposed method have been verified. In this part, we will focus on some other features of our denoising method, that is, the robustness and the real-time performance.

Firstly, as a good denoising method, it should be able to deal with various kinds of data. In order to prove this, two groups of superposed sinusoidal signals with colored noise were adopted. The sampling time for both groups was 0.001 s. The main difference between the two reference curves was that one had more sharp points while the other changed more gently, and the curves were shown respectively in Figures 9 and 10 for comparison purpose.

The first group of data with noise is given in Figure 11, where the reference curve was totally drowned. With the proposed online denoising method, the estimated curve in Figure 9 could be derived. According to the comparison with the reference curve, the original noised signal was successfully processed.

For the second group of data with noise, as was shown in Figure 12, the denoising method was applied again. The difference between the denoised result and reference values was given in Figure 10. It can be seen in the figure that the overall trend of the curve was in good accordance with the reference values, and the oscillation was because some features of the noise was reserved due to a high-dimension process model.

Secondly, we would like to discuss the real-time performance of the proposed method. In order to achieve online denoising, the algorithm should have a fast processing speed. If not, latency would exist and might affect the result. As was stated before, the method proposed in this paper was based

on Kalman filtering, which was a recursive algorithm. As long as the filtering process could finish before the new measurement was collected, the method was able to be implemented in real time. In the two simulations above, the time needed for one iteration was on average of 0.0003 s, which was far smaller than the sampling time of 0.001 s. It needs to be pointed out the difference like the subfigure shown in Figure 9 was not caused by the latency; it was mainly because of the sharp point A. The estimated points were changed with inertia, and they were then corrected to the measurement values by the recursive process. Therefore, this difference actually resulted from an estimation error other than the latency of the algorithm. In fact, the algorithm indeed performed in a real-time way as described above.

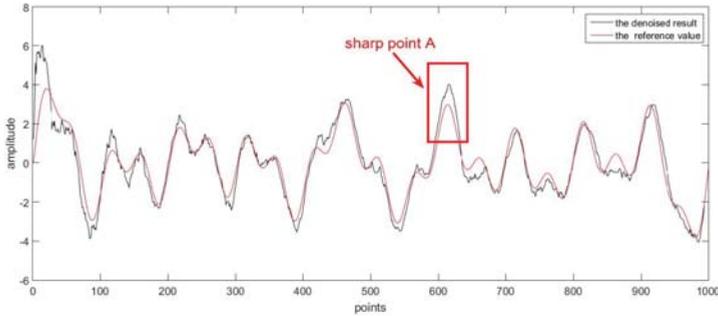


Figure 9. The denoised result and the reference value.

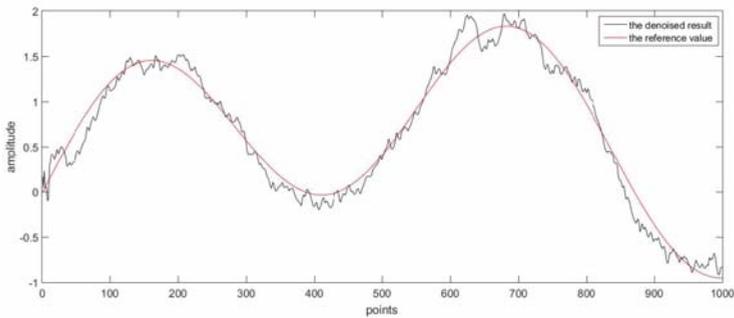


Figure 10. The reference value and the denoised result.

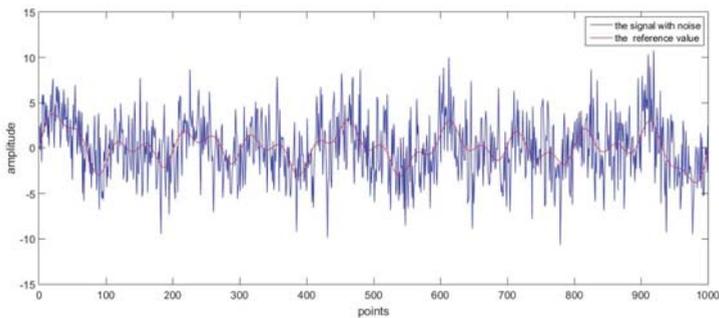


Figure 11. The signal with noise and the reference value.

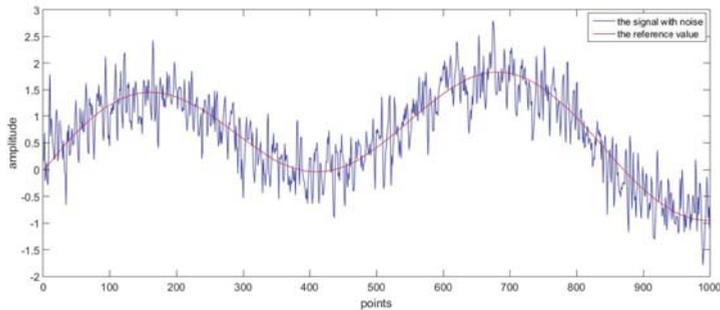


Figure 12. The reference value and the signal with noise.

5. Conclusions

A huge amount of the real-time data is collected every second around the world. However, due to the imperfect measurement and data collection mechanisms, real-time data is distorted by various types of noise and instability. Therefore, working with noisy time series is an inevitable part of any real-time data processing task and must be addressed precisely. In the past decades, the demand for real-time data analysis techniques such as the first-order exponential smoothing and Holt's exponential smoothing has grown dramatically. In this paper, we proposed an online denoising method for the real-time data with unstable fluctuation and colored noise.

This method consists of two parts within a closed loop. The first one is to estimate state based on the second-order adaptive statistics model. The other is to update the adaptive parameter in the model by the Yule–Walker algorithm. The effectiveness of method was demonstrated via an experiment, which not only processed the signals with colored noise, but also achieved a tradeoff between efficiency and accuracy. In addition, the performance of the proposed method was compared with some existing methods. Results showed that a more accurate and efficient denoising effect could be performed by employing the second-order adaptive statistics model with the Kalman filter for online denoising.

Acknowledgments: This work is partially supported by NSFC under Grant Nos. 61273002, 61673002 and 51608016, the Beijing Natural Science Foundation under Grant No. 9162002, and the Key Science and Technology Project of Beijing Municipal Education Commission of China under Grant No. KZ201510011012.

Author Contributions: Sheng-Lun Yi proposed the method and conceived and wrote the paper. Xue-Bo Jin analyzed the method. Ting-Li Su analyzed the results. Zhen-Yun Tang designed the experiment and provided the experimental data. Fa-Fa Wang, Na Xiang and Jian-Lei Kong gave advice and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, H.H.; Zhang, A.J. Improved adaptive Kalman filtering algorithm for vehicular positioning. In Proceedings of the 34th Chinese Control Conference, Hangzhou, China, 28–30 July 2015; pp. 5125–5129.
2. Chakraborty, P.; Khadivi, P.; Lewis, B.; Mahendiran, A.; Chen, J.; Butler, P.; Nsoesie, E.O.; Mekar, S.R.; Brownstein, J.S.; Marathe, M.V.; et al. Forecasting a Moving Target: Ensemble Models for ILI Case Count Predictions. In Proceedings of the SIAM International Conference on Data Mining, Philadelphia, PA, USA, 24–26 April 2014; pp. 262–270.
3. Gui, G.; Pan, H.; Lin, Z.; Li, Y.; Yuan, Z. Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. *KSCE J. Civ. Eng.* **2017**, *21*, 523–534.
4. Sapankevych, N.I.; Sankar, R. Time series prediction using support vector machines: A survey. *Comput. Intell. Mag. IEEE* **2009**, *4*, 24–38.

5. Weissman, T.; Ordentlich, E.; Seroussi, G.; Verd, S.; Weinberger, M.J. Universal discrete denoising: Known channel. *IEEE Trans. Inf. Theory* **2005**, *51*, 5–28.
6. Motta, G.; Ordentlich, E.; Ramirez, I.; Seroussi, G.; Weinberger, M.J. The idude framework for grayscale image denoising. *IEEE Trans. Image Process.* **2011**, *20*, 1–21.
7. Moon, T.; Weissman, T. Discrete denoising with shifts. *IEEE Trans. Inf. Theory* **2009**, *11*, 5284–5301.
8. Yin, A.; Zhao, L.; Gao, B.; Woo, W.L. Fast partial differential equation de-noising filter for mechanical vibration signal. *Math. Methods Appl. Sci.* **2016**, *38*, 4879–4890.
9. Pan, Q.; Zhang, D.; Dai, G.; Zhang, H. Two Denoising Methods by Wavelet Transform. *IEEE Trans. Signal Process.* **2000**, *12*, 3401–3406.
10. Ukil, A. Practical Denoising of MEG Data Using Wavelet Transform. In Proceedings of the International Conference on Neural Information Processing, Hong Kong, China, 3–6 October 2006; pp. 578–585.
11. Zou, B.; Liu, H.; Shang, Z.; Li, R. Image denoising based on wavelet transform. In Proceedings of the 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 September 2015; pp. 342–344.
12. Bhati, D.; Sharma, M.; Pachori, R.B.; Gadre, V. M. Time-frequency localized three-band biorthogonal wavelet filter bank using semidefinite relaxation and nonlinear least squares with epileptic seizure EEG signal classification. *Digit. Signal Process.* **2016**, *62*, 259–273.
13. Gao, B.; Woo, W.L.; Dlay, S.S. Single-channel source separation using EMD-subband variable regularized sparse features. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *19*, 961–976.
14. Hussein, R.; Bashirshaban, K.; El-Hag, A.H. Denoising of acoustic partial discharge signals corrupted with random noise. *IEEE Trans. Dielectr. Electr. Insul.* **2016**, *23*, 1453–1459.
15. Salih, S.K.; Aljunid, S.A.; Aljunid, S.M.; Maskon, O. Adaptive Filtering Approach for Denoising Electrocardiogram Signal Using Moving Average Filter. *J. Med. Imaging Health Inform.* **2015**, *5*, 1065–1069.
16. Mhamdi, F.; Poggi, J.; Jaidane, M. Trend extraction for seasonal time series using ensemble empirical mode decomposition. *Adv. Adapt. Data Anal.* **2011**, *3*, 363–383.
17. Alexandrov, T. A method of trend extraction using singular spectrum analysis. *Stat. J.* **2009**, *7*, 1–22.
18. Athanasopoulos, G.; Silva, A.D. Multivariate Exponential Smoothing for Forecasting Tourist Arrivals to Australia and New Zealand. Available online: <http://core.ac.uk/download/pdf/6340782.png> (accessed on 20 July 2017).
19. Seng, H. A New Approach of Brown's Double Exponential Smoothing Method in Time Series Analysis. *Balk. J. Electr. Comput. Eng.* **2016**, *4*, 75–78.
20. Jere, S.; Siyanga, M. Forecasting Inflation Rate of Zambia Using Holt's Exponential Smoothing. *Open J. Stat.* **2016**, *6*, 363–372.
21. Goh, B.M.K.; Lim, H.S.; Tan, A.W.C. Exponential Myriad Smoothing Algorithm for Robust Signal Processing in α -Stable Noise Environments. *Circuits Syst. Signal Process.* **2017**, *8*, 1–14.
22. Wang, D.; Lv, H.; Wu, J. In-flight initial alignment for small UAV MEMS-based navigation via adaptive unscented Kalman filtering approach. *Aerosp. Sci. Technol.* **2017**, *61*, 73–84.
23. Zhou, S.; Hu, P.; Li, K.; Liu, Y. A New Target Tracking Scheme Based on Improved Mean Shift and Adaptive Kalman Filter. *Int. J. Adv. Comput. Technol.* **2012**, *4*, 291–301.
24. Rivas, M.; Xie, S.; Su, D. An adaptive wideband beamformer using Kalman filter with spatial signal processing characteristics. In Proceedings of the International Conference on Advanced Communication Technology, Seoul, Korea, 13–16 February 2011; pp. 321–325.
25. Sinharay, A.; Pal, A.; Bhowmick, B.; A Kalman Filter Based Approach to De-noise the Stereo Vision Based Pedestrian Position Estimation. In Proceedings of the 13th International Conference on Computer Modelling and Simulation, Cambridge, UK, 30 March–1 April 2011; pp. 110–115.
26. Huang, Z.K.; Liu, D.H.; Zhang, X.W.; Hou, L.Y. Application of Kalman Filtering for Natural Gray Image Denoising. *Adv. Mater. Res.* **2011**, *187*, 92–96.
27. Rosa, G.D.; Petukhov, Y. A Kalman Filter approach for track reconstruction in a neutrino telescope. *Nucl. Instrum. Methods Phys. Res.* **2013**, *725*, 118–121.
28. Miao, B.B.; Dou, C.; Jin, X.B. Main Trend Extraction Based on Irregular Sampling Estimation and Its Application in Storage Volume of Internet Data Center. *Comput. Intell. Neurosci.* **2016**, *2016*, 9328062.

29. Jin, X.B.; D, J.J.; Bao, J. Maneuvering target tracking by adaptive statistics model. *J. China Univ. Posts Telecommun.* **2013**, *20*, 108–114.
30. Tang, Z.Y.; Ma, H.; Guo, J.; Xie, Y.; Li, Z. Experimental research on the propagation of plastic hinge 305 length for multi-scale reinforced concrete columns under cyclic loading. *Earthq. Struct.* **2016**, *11*, 823–840.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

A Time-Space Domain Information Fusion Method for Specific Emitter Identification Based on Dempster–Shafer Evidence Theory

Wen Jiang ^{1,*}, Ying Cao ¹, Lin Yang ^{2,*} and Zichang He ¹

¹ School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China; caoyingnwpu@hotmail.com (Y.C.); hezichang@mail.nwpu.edu.cn (Z.H.)

² China Equipment System Engineering Company, Beijing 100039, China

* Correspondence: jiangwen@nwpu.edu.cn (W.J.); yanglin61s@sina.com (L.Y.); Tel.: +86-29-8843-1267 (W.J.)

Received: 23 July 2017; Accepted: 24 August 2017; Published: 28 August 2017

Abstract: Specific emitter identification plays an important role in contemporary military affairs. However, most of the existing specific emitter identification methods haven't taken into account the processing of uncertain information. Therefore, this paper proposes a time–space domain information fusion method based on Dempster–Shafer evidence theory, which has the ability to deal with uncertain information in the process of specific emitter identification. In this paper, radars will generate a group of evidence respectively based on the information they obtained, and our main task is to fuse the multiple groups of evidence to get a reasonable result. Within the framework of recursive centralized fusion model, the proposed method incorporates a correlation coefficient, which measures the relevance between evidence and a quantum mechanical approach, which is based on the parameters of radar itself. The simulation results of an illustrative example demonstrate that the proposed method can effectively deal with uncertain information and get a reasonable recognition result.

Keywords: Dempster–Shafer evidence theory; specific emitter identification; time–space domain information fusion; quantum mechanical approach; correlation coefficient; recursive centralized model

1. Introduction

Information fusion, also known as multi-sensor data fusion, is a process of synthesizing data or information obtained from multiple sources to achieve a certain purpose, which has been widely used in many fields, such as detection [1–3], recognition [4,5], tracking [6–9], image processing [10–12], fault diagnosis [13–17], and gender profiling [18]. In recent years, the rapid development of sensor technology and computer technology has greatly promoted the research of information fusion technology [19–22]. As an uncertain information processing method that satisfies the weaker conditions than traditional Bayesian probability theory, Dempster–Shafer evidence theory [23,24] is widely adopted by scholars in information fusion [25–29].

Information fusion incorporates space-domain fusion and time-domain fusion. In space domain, it is a process of fusing multiple sets of evidence generated by multiple sensors in space. Since the credibility of a body of evidence generated by a single sensor can not be determined, the integration of multi-sensor information in space-domain is beneficial to people in order to make more accurate judgements than a single sensor. However, it doesn't take into account the changes in evidence over time. While in practical applications, the information obtained by different measurement periods may change due to the interference of other factors, and at this time, the time-domain fusion, which is a fusion process of evidence generated at different moments with the properties of dynamic, sequential and real-time, is very important. In general, the time-domain and the space-domain are both

indispensable in the process of the information fusion. At present, there are a variety of space-domain information fusion results [30–36], and the time domain information fusion has also been gradually adopted by scholars [37–39]. Hong and Lynch [40] proposed the space-time information fusion models based on evidence theory, including recursive centralized fusion model, recursive distribution without feedback fusion model and recursive distribution with feedback fusion model. In addition, Hong [41] made profound analysis and comparison of these three models. However, they didn't give these three models specific application background. In this paper, the recursive centralized model is applied to specific emitter identification.

Radar plays an important role in contemporary military affairs. However, the modern electromagnetic environment is complex since there are signals from a number of different emitters present, and signals from the same emitter are noisy or their parameters (features) are not measured with a great accuracy [42]. Therefore, how to distinguish the correct information from the acquired signal and obtain a reasonable result is a difficult problem. This field is called radar target recognition (RTR), which is widely used in command automation, identification friend or foe and intelligence acquisition and meets the needs of modern high-tech warfare and helps to realize the intelligence and informatization of radar. In RTR, there is a process of source (target) recognition and classification and identification. Recognition concerns type classification, while identification focuses recognition on particular copies of the same radar (target) type [43]. The identification process is more specialized as it requires methods based on the analysis of distinctive features. These features are then identified based on the information obtained. This is called specific emitter identification (SEI). The main task of SEI is to identify radar emission sources. Although there is much research about SEI [43–45], most of them do not consider the processing of uncertain information, which means that the identification method in the references [43–45] start from the point of view of signal processing. However, Dempster–Shafer evidence theory holds that we can only say radar is more likely to belong to which type, not to say which radar definitely belongs to which type. Therefore, in this paper, we assume that radar will give a set of evidence about the specific identity of the unknown target after signal analysis. Our main task is to fuse multiple sets of evidence to identify the identity of an unknown target.

He and Jiang [46] proposed a quantum mechanical approach based on Dempster–Shafer evidence theory, which considered work performance of radars themselves to model reliability of sensor reports, while most of the previous methods only considered the interrelationships between evidence generated by sensors. However, the quantum mechanical approach [46] failed to deal with the impact of time factors on the fusion results effectively. Therefore, in this paper, a time–space domain information fusion method for SEI based on Dempster–Shafer evidence theory is proposed. At first, as the distance between the radar and the target will change over time, the fusion of time-domain evidence is joined. Then, the proposed method adopts correlation coefficient [47] and quantum mechanical approach to generate weighting factors (weights). At last, within the framework of recursive centralized model, the evidence is combined based on Dempster's combination rule. The space-time domain information fusion method of SEI proposed in this paper takes full account of the influence of time factors on the fusion results so that it has a strong dynamic nature. In addition, it incorporates the interrelationship between evidence that is measured by correlation coefficient and the impact of the radar's own performance, which considered by a quantum mechanical approach at the same time, so its management of conflict information is more comprehensive.

The structure of this paper is as follows: in Section 2, the relevant background knowledge is introduced, including Dempster–Shafer evidence theory, correlation coefficient and recursive centralized fusion model. Section 3 describes the quantum mechanical approach and discuss it. Section 4 introduces the specific steps of the proposed method. In addition, the proposed method is simulated in Section 5. Section 6 presents the conclusions.

2. Preliminaries

2.1. Dempster–Shafer Evidence Theory

Dempster–Shafer evidence theory (D–S evidence theory) [23,24] is more extensive than traditional Bayesian probability with it satisfying the weaker prior conditions, and this concept contributes to its good performance in handling the uncertainty information [48–50]. It is briefly introduced as follows.

Definition 1. Suppose Θ is a nonempty finite set composed of n exhaustive and exclusive elements, and it satisfies:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_N\}, \quad (1)$$

where set Θ is called a frame of discernment. The power set of Θ , 2^Θ , is indicated as:

$$2^\Theta = \{\emptyset, \{\theta_1\}, \dots, \{\theta_N\}, \{\theta_1, \theta_2\}, \dots, \{\theta_1, \theta_2, \dots, \theta_i\}, \dots, \Theta\}. \quad (2)$$

Definition 2. A mass function is a mapping m from 2^Θ to $[0, 1]$, formally defined as:

$$m : 2^\Theta \rightarrow [0, 1], \quad (3)$$

and it satisfies:

$$m(\emptyset) = 0, \quad (4)$$

$$\sum_{A \in 2^\Theta} m(A) = 1. \quad (5)$$

The function m is called Basic Probability Assignment (BPA), and $m(A)$ indicates the degree of trust in proposition A . If $m(A) > 0$, A is called a focal element of Θ . If $m(A) = 0$, it means that the proposition is totally lacks belief. In addition, a value between $[0, 1]$ indicates partial belief.

Definition 3. In D–S theory, the combination is denoted as \oplus . Supposing that there are two BPAs denoted by m_1 and m_2 , the two BPAs' combination with the Dempster's combination rule is then formulated as follows:

$$m_{\oplus}(A) = \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - k}, \quad A \neq \emptyset, \quad (6)$$

$$m_{\oplus}(\emptyset) = 0, \quad (7)$$

where

$$k = m(\emptyset) = \sum_{B \cap C = \emptyset} m_1(B)m_2(C). \quad (8)$$

Essentially, k reflects the degree of conflict among evidence. In D–S theory, if $k = 0$, we say that the two evidence is fully compatible with each other. On the contrary, if $k = 1$, we say that the two pieces of evidence are in total conflict with each other. In addition, the Dempster's combination rule is commutative and associative, which ensures that the fusion result has nothing to do with the order of the fusion process, and it essentially assigns the mass of empty set to each set by the use of the normalization. It has two characteristics:

- Pieces of mutual support evidence are reinforced.
- Pieces of conflict evidence weaken each other.

However, the direct use of the Dempster's combination rule to integrate the evidence generated by information sources may produce counterintuitive results [51].

2.2. Correlation Coefficient of Belief Function

How to measure conflict between evidence is still an open issue. In classical D–S evidence theory, the mass of empty set k is used to reflect the degree of conflict among evidence. Actually, k defines the compatibility between the evidence. Jousselme [52] proposed evidence distance to measure conflict from the perspective of the similarity between evidence. Liu [53] proposed a two-dimensional measure that contains k and evidence distance and considers the compatibility and similarity at the same time. In addition, many other scholars studied the problem [54–57]. In this section, we will introduce a new correlation coefficient [47], which measures the relevance between evidence and contains compatibility and similarity based on our previous work.

Definition 4. A discernment frame is assumed with n elements, and two pieces of evidence are denoted by m_1 and m_2 . The correlation coefficient is defined as Equation (9):

$$r_{BPA}(m_1, m_2) = \frac{c(m_1, m_2)}{\sqrt{c(m_1, m_1) \cdot c(m_2, m_2)}}, \tag{9}$$

where

$$c(m_1, m_2) = \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} m_1(A_i) m_2(A_j) \frac{|A_i \cap A_j|}{|A_i \cup A_j|}, \tag{10}$$

where $i, j = 1, 2, \dots, 2^n$; A_i and A_j are the elements of BPAs m_1 and m_2 ; In addition, $|\cdot|$ is the cardinality of a subset. Actually, $c(m_1, m_2)$ shows the degree of correlation.

The correlation coefficient r_{BPA} measures the relevance between evidence m_1 and m_2 . The larger the r_{BPA} , the greater the relevance between the evidence, and the lower the degree of conflict. In extreme cases, we can say that evidence is in total conflict if $r_{BPA} = 0$ and in the total absence of conflict if $r_{BPA} = 1$.

2.3. Recursive Centralized Model

Hong and Lynch [40] studied time–space information fusion model based on D–S evidence theory. Until now, there are three basic models: recursive centralized fusion model, recursive distribution without feedback fusion model and recursive distribution with feedback fusion model [40]. The recursive centralized fusion model that is used in our method can realize real-time fusion and high precision of data processing.

The recursive centralized fusion process is shown in Figure 1. At moment t , it first fuses N evidence generated by N sensors, and then fuses it with cumulative information denoted as $m(t - 1)$. The fused result could include total target identification information at the moment t .

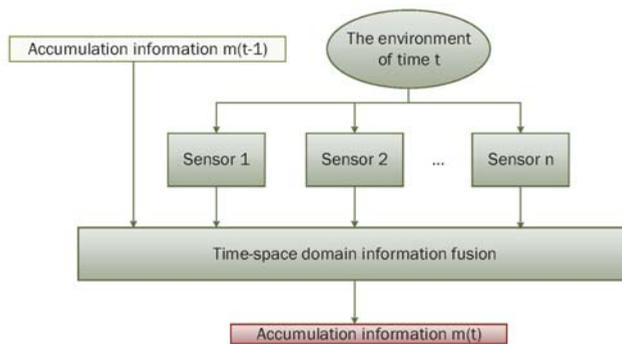


Figure 1. Recursive centralized model.

2.4. Radar Working Principle

Radar is a device that detects and measures information of targets by transmitting electromagnetic waves and receiving echoes. The specific use and structure of the various radars are different, but the basic form is consistent, including: transmitter, transmitting antenna, receiver, receiving antenna, processing part and display. In addition, there is also power equipment, data entry equipment, anti-jamming equipment and other auxiliary equipment. Generally speaking, radars are divided into active radars and passive radars. Their main difference is that the active radar transmits and receives signals, while the passive radar only receives signals. In this paper, we adopt passive radar. The process of radar information processing is shown in Figure 2. In this paper, radar generates a piece of evidence based on the information obtained that include: relative position, speed, heading, closest point of approach (CPA) and distance to CPA.



Figure 2. Working process of radar.

In general, parameters related to the performance of the radar are shown in Definitions 5 and Equation (12).

Definition 5. Since the passive radar only receives the signal, the signal is one-way. The one-way signal strength received by radar is defined as follows:

$$P_r = \frac{P_t G_t G_r \sigma \lambda^2}{(4\pi x)^2}, \quad (11)$$

where P_t defines the transmission ability of radar, with G_t the gain of target antenna, G_r the gain of radar antenna, σ the radar cross section (including the geometric cross-sectional area, reflection coefficient and direction coefficient) factor, r the wavelength, and x the distance between the target object and the radar.

Definition 6. The maximal reconnaissance distance is defined as follows:

$$x_r = \left[\frac{P_t G_t G_r \sigma \lambda^2}{(4\pi)^2 P_{r \min}} \right]^{\frac{1}{2}}, \quad (12)$$

where $P_{r \min}$ is the minimum signal strength radar can receive.

If the distance between radar and target beyond x_r , the signal strength received by radar is too small for radar to detect it properly.

3. Existing Quantum Mechanical Approach

He and Jiang [46] proposed a quantum mechanical method that considers the performance of the radar sensor itself to assign weights among evidence based on the link of uncertainty between quantum mechanics and D-S evidence theory. In this section, we will introduce this method and discuss it.

In classical mechanics, it is generally believed that state of a particle is determined at some points in time where the state of the particle is usually expressed by the coordinates and momentum of the particle. However, the position and momentum of the particle can not be determined at the same time according to the Heisenberg uncertainty principle in quantum mechanics, and they can only be in the form of probability. Therefore, we can only say that particles are more likely to be in which state, but can not be sure the particles is in one state as Figure 3 shows.

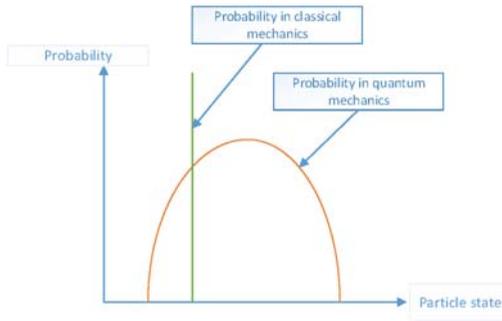


Figure 3. Particle state and probability.

In classical mechanics, if the state of the particle is known at a given moment, the state of the particle can be obtained at any time after the particle is determined according to its equation of motion. Similarly, in quantum mechanics, when the state of the microscopic particles is known at some point, the state of motion of the particles can also be determined by an equation, which is called Schrodinger equation.

Definition 7. One state becomes stationary when the energy of a system in which the particle located is a certain value. The equation of the Schrodinger equation is as follows:

$$-\frac{\hbar^2}{2m}\nabla^2\Psi + U(\vec{r})\Psi = E\Psi, \tag{13}$$

where E is total energy (also called the overall state) of the system, Ψ represents wave function, $-\frac{\hbar^2}{2m}$ is kinetic energy, $U(\vec{r})$ is Potential energy and ∇ is Laplacian operator. $-\frac{\hbar^2}{2m}\nabla^2 + U(\vec{r})$ is called Hamiltonian operator.

The wave function depicts the quantum state of the system, which is proportional to the probability that the intensity of a point in the space (the square of the absolute value of the amplitude) and the probability of finding the particle at that point. According to this interpretation, the wave describing the particle is a probability wave. Actually, it is essentially a probability distribution curve that represents the probability that the particle is in the corresponding state. According to this property, Bolotin [58] solved the Schrodinger equation with a membership curve of the problem solution, and then the optimal solution of the problem is obtained. In Bolotin’s approach, the most important step is to set up quasi-potential function and then put it into the Schrodinger equation and solve it. Similarly, since the performance of the radar is related to signal-to-noise ratio (SNR), and SNR is related to the received signal strength, we regard the signal strength as quasi-potential function, shown in Equation (14):

$$U(x) = \begin{cases} \frac{\gamma}{x^2} & 0 < x < x_r \\ \infty & x \leq 0, x > x_r, \end{cases} \tag{14}$$

where

$$\gamma = \frac{P_t G_t G_r \sigma \lambda^2}{(4\pi)^2}. \tag{15}$$

γ corresponds to the parameters of Equation (11). In addition, it is impossible for a particle to penetrate the well wall if it is within a infinite well potential, so $\Psi(x) = 0$ when the distance beyond the maximal reconnaissance distance.

In addition, then, we put $U(x)$ into the Schrodinger equation and solve it, we can get the wave function. In addition, with it, the probability distribution curve $P(x)$ can be obtained in Equation (16):

$$P(x) = |\Psi(x)|^2 \propto x \left[J_\alpha\left(\frac{\sqrt{L}}{c}\right) + Y_\alpha\left(\frac{\sqrt{L}}{c}\right) \right]^2, \tag{16}$$

where J_α and Y_α are Bessel functions of the first kind and the second kind, respectively [59], and $\alpha = \frac{1}{2} \sqrt{\frac{c^2 - 4\gamma}{c^2}}$, which denotes their order.

Normalizing probability distribution, we can obtain the relationship curve between radar working distance and membership, shown in Figure 4. The higher the membership degree is when the radar is at a certain working distance, the better performance of the radar, and the more reliable the evidence generated by the radar is. Deng [30] proposed a method of the distribution of weights based on evidence distance that measures the similarity between evidence. Similarly, weights among evidence can be assigned based on the memberships we obtained that show the performance of multiple radars themselves. The process of the quantum mechanical approach is shown in Figure 5.

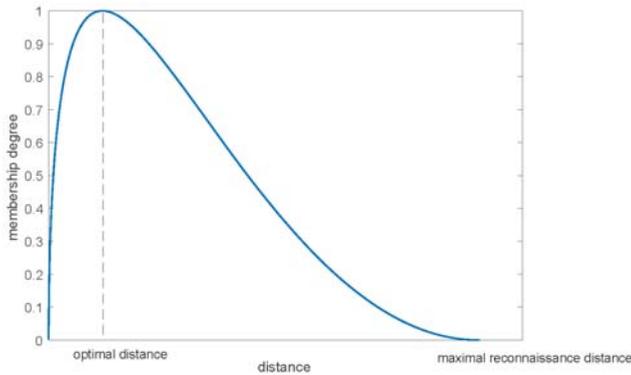


Figure 4. Radar’s working distance and membership degree.

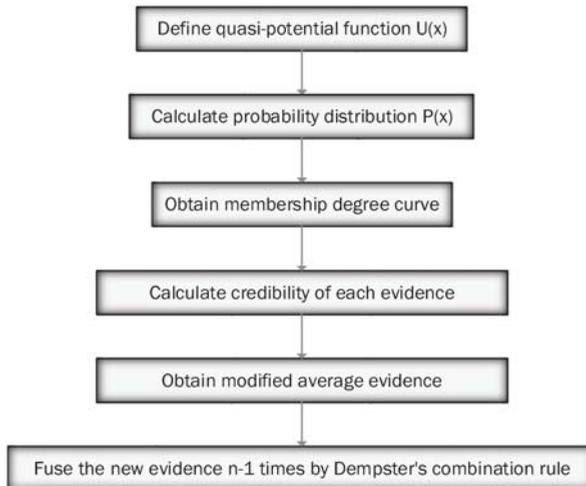


Figure 5. The process of the existing quantum mechanical approach.

4. Proposed Method to Realize SEI

He and Jiang [46] opened the way to consider the performance of radar work, but their method failed to effectively deal with the impact of time on the entire process. In this section, a time–space domain information fusion method for SEI will be proposed, which considers both the relationship between evidence and performance of radars themselves based on evidence theory and the existing quantum mechanical approach.

Most of the current methods of information fusion are carried out in space-domain that is based on multi-sensor fusion. However, in real applications, the information obtained by a single measurement cycle has a certain contingency as it may be interfered with by electromagnetic waves, weather and other factors. In addition, this kind of contingency can be prevented by time-domain information fusion. In addition, the system of time-domain fusion has properties of real-time, sequence and dynamic, which can achieve inheritance and update of the integration. In this paper, the evidence and distance between radars and target will change over time. In addition, the confidence of evidence generated by radar sensors may then be different at different moments if we select multiple sets of evidence at multiple moments.

Now, supposing that there are n radars and t measurement cycles, the specific steps of the method are shown as follows:

- Step 1 Calculate the correlation coefficient, and then get the weighting factors based on evidence. r_{kj} is the correlation coefficient between evidence k and j , and then we can get a matrix:

$$R = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1n} \\ r_{21} & 1 & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & 1 \end{bmatrix}. \tag{17}$$

The degree of trust of a piece of evidence can be defined as:

$$\lambda_j = \sum_{k=1, k \neq j}^n r_{kj}. \tag{18}$$

However, since we are in the system of time-domain fusion where evidence is different at different moments, λ_j is different at different moments. Therefore, it needs to be defined as λ_{ij} , which represents the support degree of radar j at moment i .

- Step 2 Calculate membership degree curve of each radar by quantum mechanical approach based on the performance parameters of given radars, and then get the membership degree ω_{ij} of the corresponding working distance at each moment.
- Step 3 Calculate the final weights μ_{ij} , which is the final weight of a piece of evidence generated by radar j at moment i by summing and normalizing:

$$\mu_{ij} = \frac{\omega_{ij} \cdot \lambda_{ij}}{\sum_{j=1}^n (\omega_{ij} \cdot \lambda_{ij})}. \tag{19}$$

- Step 4 The credibility of each evidence μ_{ij} is regarded as a weighting factor to weight the evidence at moment i , and then we get a new piece of modified average evidence (MAE):

$$MAE = \sum_{j=1}^n (\mu_{ij} \cdot m_i). \tag{20}$$

Step 5 Get the fusion results of the current moment by fusing the new piece of evidence $n - 1$ times, and then fuse the fusion results of the current moment and the previous moment by Dempster’s combination rule. This is based on the recursive centralized model.

The process of the proposed method is shown in Figure 6. In this method, the parameter λ_{ij} represents the weight that comes from the perspective of relationships between evidence. While ω_{ij} is the weight on the radar itself’s perspective. We find a balance denoted by μ_{ij} between these two kinds of weights when we multiply the two parameters. Therefore, the proposed method not only considers the data’s relationship, but also the performance of the sensor itself, while most of the existing methods only consider one issue. In the next section, the proposed method will be simulated and compared with other existing methods.

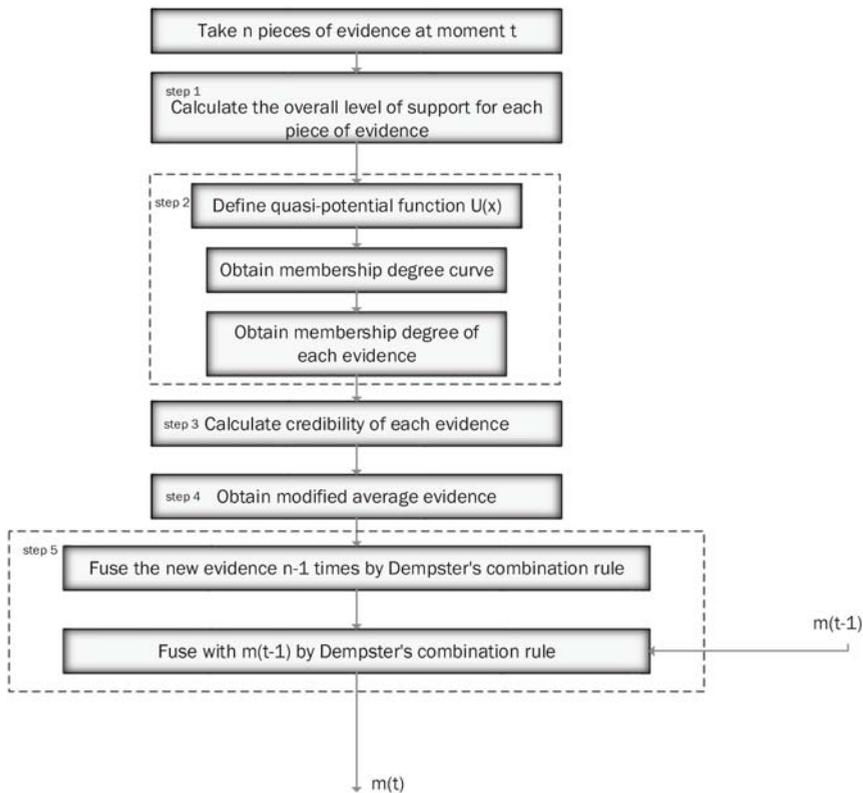


Figure 6. The process of the proposed method.

5. Case Study

In this section, we simulate the data fusion process of radar practically by fusing the assumed BPAs, and several simulation results are presented here to test the performance of the proposed method of SEI. Our analysis process is carried out from three aspects:

- Comparing the proposed method with Dempster, Deng et al. [30] and He and Jiang [46] when the target moves in a straight line.
- Comparing the proposed method with the other three methods when the target moves along a curve.

- Using self performance comparison when the number of radars changes and when we only sample one moment, the target moves in a straight line at five moments.

Assume frame of discernment is: $\Theta = \{A, B, C\}$, which means that the identity of the unknown target is one of the three elements. In this section, we assume that "A" is the correct identification result. There are five radars distributed in space and they are likely to be disturbed in the region to be monitored because of weather, noise or some other factors. Their relative positions with the target are shown in Figure 7, and the target travels along a straight line to radar 3. Now, we select five moments denoted by t_1, t_2, t_3, t_4 and t_5 , and the BPAs generated at different moments by the five radars based on information received are shown in Table 1.

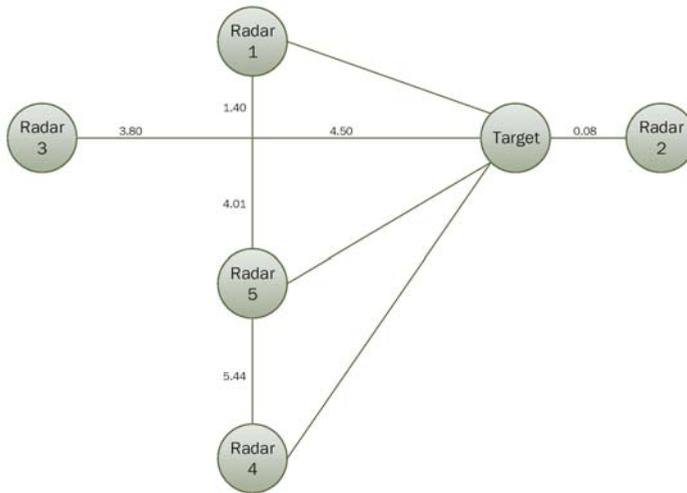


Figure 7. Relative position of five radars with the target when the target moves along a straight line.

Table 1. Basic Probability Assignments generated at different moments by the five radars.

		m_1	m_2	m_3	m_4	m_5
t_1	{A}	0.55	0.5	0.65	0.5	0.5
	{B}	0.25	0.35	0.2	0.2	0.15
	{C}	0.1	0.05	0.1	0.2	0.1
	{A, B, C}	0.1	0.1	0.05	0.1	0.25
t_2	{A}	0.6	0	0.5	0.5	0.55
	{B}	0.15	0.95	0.3	0.1	0.25
	{C}	0.2	0.05	0.1	0.15	0.1
	{A, B, C}	0.05	0	0.1	0.25	0.1
t_3	{A}	0.6	0.6	0	0.1	0.55
	{B}	0.1	0.2	0.6	0.7	0.35
	{C}	0.1	0.1	0.15	0.1	0.05
	{A, B, C}	0.2	0.1	0.25	0.1	0.05
t_4	{A}	0.55	0.4	0.55	0.5	0
	{B}	0.2	0.3	0.2	0.1	0.5
	{C}	0.1	0.2	0.2	0.2	0.5
	{A, B, C}	0.15	0.1	0.05	0.2	0
t_5	{A}	0.5	0.5	0.6	0.55	0.4
	{B}	0.2	0.2	0.2	0.35	0.4
	{C}	0.1	0.2	0.1	0.1	0.1
	{A, B, C}	0.2	0.1	0.1	0	0.1

The maximum reconnaissance distances for five radars are 9.1, 11.0, 12.3, 14.0 and 15.1, and performance parameters c , L and γ are shown in Table 2, where c is used to control the width of the curve, L is the inherent state of the whole system, and it is proportional to $P_{r\min}$. γ is shown in Equation (15), representing the radar's own parameters. Five curves can be obtained that describe the relationship between distance and membership degree (shown in Figure 8). The distance between radar and target has been changing over time, so the radar's membership degree at different moments are different. Distances between five radars and target at five moments are shown in Table 3. According to the distances between the radar and the target, we can get the membership degree (that is, the performance of the five radars). In addition, the final results can then be calculated, which are shown in Table 4. The trend of the results at five moments is shown in Figure 9.

Table 2. The value of performance parameters.

	c	L	γ
radar a	16	0.64	47.25
radar b	16	0.41	48.13
radar c	16	0.31	49.75
radar d	16	0.17	53.01
radar e	16	0.20	50.49

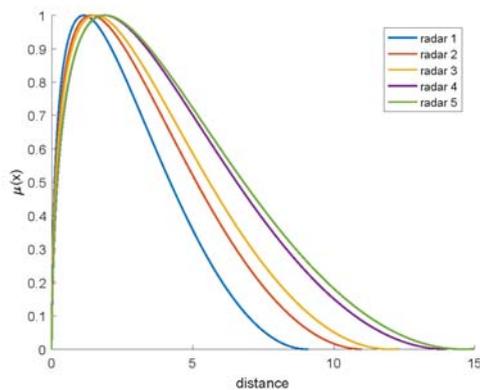


Figure 8. The relationship between distance and membership degree.

In the process of simulation above, the SEI method proposed in this paper was compared with three existing methods, including Dempster's combination rule, Deng's combination rule [30], and He and Jiang's method [46]. Firstly, at moment t_1 , all of the radars are working properly. In this case, all four of the methods yield reasonable results. Secondly, the classical Dempster's combination rule has a characteristic named "one-vote veto", which means that if one piece out of all of the evidence gives the proposition "A" complete negation, the final fusion result will completely negate "A" (like the time t_2 , and t_4). We can see that the proposed method of this paper solved this problem. Thirdly, He and Jiang's quantum mechanical approach [46] opened up the precedence of the distribution of weights based on the radar's own performance. However, they didn't take the impact of previous results on the current moment and relationship among evidence into account. In this case, it is likely to produce unreasonable results if a disturbed radar at one moment happens to work at the degree of membership of 1, but other radars' memberships are small, such as the evidence generated at t_2 . In addition, at this moment, interrelationship between evidence (correlation coefficient [47] applied in this paper) needs to be reconciled, and our method

solves this problem. Finally, Deng [30]’s rule only considered the interrelationship among evidence and its strategy’s ability of anti-interference is poor. As in the case of t_3 , when there are two radars that are interfered with and produce the wrong BPAs, Deng’s result will have a higher degree of uncertainty or even a false result. In this paper, we introduce time-domain fusion that takes the effect of the previous result at the current moment into account. In this case, even if the trust in “A” (the correct result) is not very high due to the interference in the current time, “A” will be still have a high degree of trust after recursive centralized fusion if the previous result supports “A”. This reflects the ability of anti-interference and the integration of the results of the succession and update.

Table 3. Distances between five radars and target at five moments when the target moves along a straight line.

	Radar 1	Radar 2	Radar 3	Radar 4	Radar 5
t_1	4.7127	0.08	8.3	10.4667	6.0274
t_2	3.31	1.58	6.8	9.92	5.01
t_3	2.0447	3.08	5.3	9.5737	4.2837
t_4	1.4	4.58	3.8	9.45	4.01
t_5	2.0447	6.08	1.3	9.5737	4.2837

Table 4. Simulation results of four methods when the target moves along a straight line.

		t_1	t_2	t_3	t_4	t_5
<i>Dempster’s rule</i>	{A}	0.9595	0	0.4022	0	0.9317
	{B}	0.0368	0.9790	0.5863	0.5863	0.0613
	{C}	0.0036	0.0210	0.0109	0.4167	0.0068
<i>Deng’s rule</i>	{A}	0.9555	0.8849	0.5142	0.8316	0.9209
	{B}	0.0395	0.1055	0.4764	0.0972	0.0735
	{C}	0.0048	0.0093	0.0088	0.0710	0.0055
<i>He and Jiang’s method</i>	{A}	0.9249	0.2538	0.8132	0.6749	0.9203
	{B}	0.0712	0.7435	0.1791	0.2087	0.0733
	{C}	0.0035	0.0026	0.0071	0.1163	0.0061
<i>Proposed method</i>	{A}	0.9243	0.9703	0.9944	0.9989	0.9995
	{B}	0.0718	0.0294	0.0055	0.0009	0.0002
	{C}	0.0036	0.0002	0.0001	0.0001	0.0002

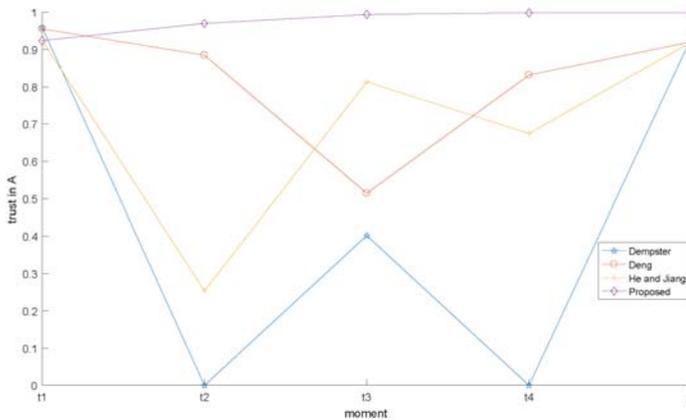


Figure 9. The trend of the results at five moments when the target moves along a straight line.

Now, we change the movement trajectory of the target, and then the relative position of five radars with the target changes with it. However, the performance parameters of the five radars and BPAs in five moments do not change. Therefore, we compare the proposed method with the other three methods. The simulation process is shown below. We assume that the target moves along a circular path with a radius of 1 as shown by the dotted line in Figure 10 and the spatial location of five radars is the same as with Figure 7. Starting from the moment t_1 , the unknown target moves a quarter of the circle at every moment, and back to the origin at moment t_5 . Distances between five radars and the unknown target at five moments are shown in Table 5. According to the distances between the radars and the target, we can get the membership degree with the basis of Figure 8. In addition, the final results of proposed method and the other three methods can then be calculated, which is shown in Table 6. The trend of the results at five moments is shown in Figure 11.

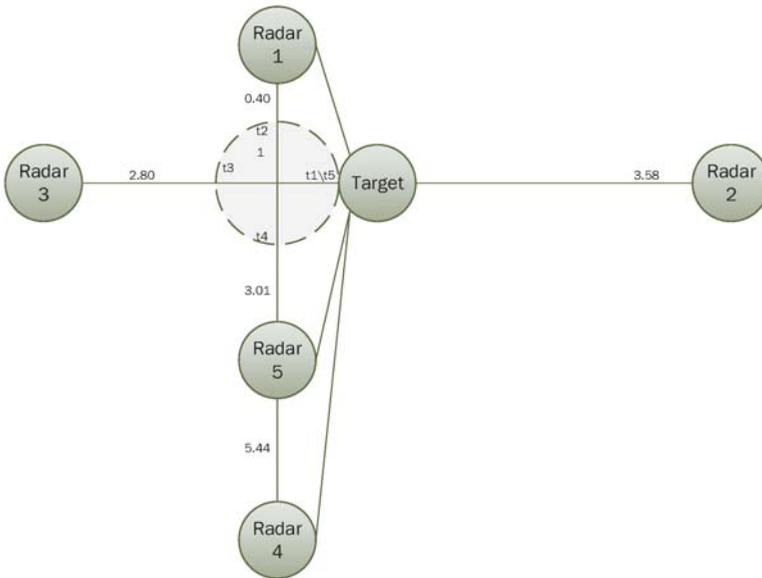


Figure 10. Relative position of five radars with the target when the target moves along curves.

Table 5. Distance between five radars and the target at five moments when the target moves along curves.

	Radar 1	Radar 2	Radar 3	Radar 4	Radar 5
t_1	1.72	3.58	4.8	9.5	4.13
t_2	0.4	4.69	3.93	10.45	5.01
t_3	1.72	5.58	2.8	9.5	4.13
t_4	2.4	4.68	3.93	8.45	3.01
t_5	1.72	3.58	4.8	9.5	4.13

Since Dempster and Deng [30]’s combination results are only relevant to the data given to the evidence, but not to the relative position of the radar with the target, their combination results at five moments are the same as Table 4 if we change the movement trajectory of the target. When we change the movement trajectory of the target, the fusion results also change since the performance of He and Jiang’s method [46] has a great relationship with distance between target and radar. The proposed method is in the same situation. From Figure 11, we can see that the uncertainty of the fusion results of

the proposed method is always low but fluctuates with He and Jiang’s method [46] in five moments. Why does this happen? As we discussed before, the three fusion methods that we adopted to compare with our method only consider the BPAs of the current moment. However, in practical applications, it is difficult for us to decide which moment to determine. The proposed method takes the effect of the previous fusion result into account. Therefore, in general, the robustness of the proposed method is better than the other three methods.

Table 6. Simulation results when the target moves along curves.

		t_1	t_2	t_3	t_4	t_5
<i>Dempster's rule</i>	{A}	0.9595	0	0.4022	0	0.9317
	{B}	0.0368	0.9790	0.5863	0.5863	0.0613
	{C}	0.0036	0.0210	0.0109	0.4167	0.0068
<i>Deng's rule</i>	{A}	0.9555	0.8849	0.5142	0.8316	0.9209
	{B}	0.0395	0.1055	0.4764	0.0972	0.0735
	{C}	0.0048	0.0093	0.0088	0.0710	0.0055
<i>He and Jiang's method</i>	{A}	0.9516	0.8808	0.5571	8429	0.9112
	{B}	0.0446	0.1099	0.4732	0.0914	0.0799
	{C}	0.0036	0.0091	0.0085	0.0654	0.0086
<i>Proposed method</i>	{A}	0.9518	0.9942	0.9972	0.9995	0.9999
	{B}	0.0444	0.0055	0.0026	0.0004	0.00005
	{C}	0.0033	0.0003	0.0002	0.0001	0.00005

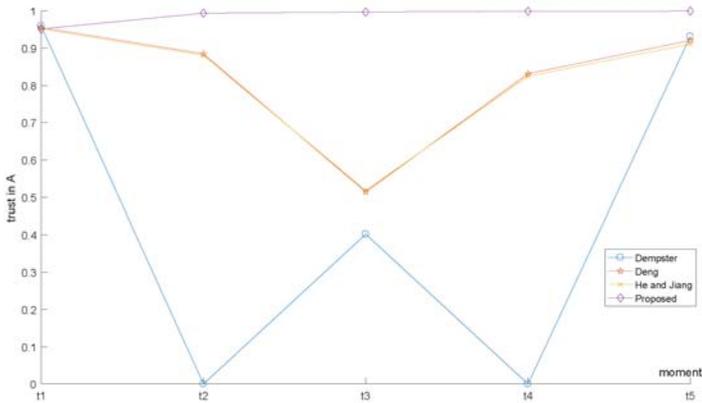


Figure 11. The trend of the results at five moments when the target moves along curves.

Now, we analyze the effect of variations in radar number on fusion results. BPAs of five radars at five moments do not change. From the value of BPAs, we can see that five radars are working properly since they all support the proposition “A” at t_1 , while there are some radars not working properly at t_2, t_3, t_4 and t_5 . Since the effect of variations in radar number on the performance of the proposed method in this simulation needed to be determined, we only use self performance comparison of the proposed method at five moments. In addition, the relative positions of the five radars with targets is the same as Figure 7. When a different number of radar sensors are used in our identification system, the fusion results are shown in Table 7. In Table 7, we assume that only the BPAs of the current time are considered in order to show the importance of sampling at multiple moments. The trend of fusion results from different numbers of radars are shown in Figure 12.

Table 7. Simulation results of the proposed method when the number of radars changes.

Number of Radars		1	2	3	4	5
t_1	{A}	0.55	0.6849	0.8457	0.9136	0.9243
	{B}	0.25	0.2586	0.1388	0.0807	0.0718
	{C}	0.1	0.0387	0.0135	0.0054	0.0036
t_2	{A}	0.6	0.3442	0.6311	0.7803	0.8871
	{B}	0.1	0.5897	0.3354	0.2024	0.1073
	{C}	0.1	0.0641	0.0324	0.0171	0.0055
t_3	{A}	0.55	0.8079	0.6494	0.4675	0.6653
	{B}	0.2	0.0883	0.2668	0.4789	0.3235
	{C}	0.1	0.0629	0.0632	0.0457	0.0103
t_4	{A}	0.5	0.6554	0.8140	0.8988	0.7822
	{B}	0.2	0.2144	0.1215	0.0652	0.1384
	{C}	0.1	0.1001	0.0607	0.0350	0.0792
t_5	{A}	0.55	0.6785	0.8371	0.9129	0.9125
	{B}	0.25	0.1711	0.1048	0.0648	0.0786
	{C}	0.1	0.1097	0.0501	0.0209	0.0086

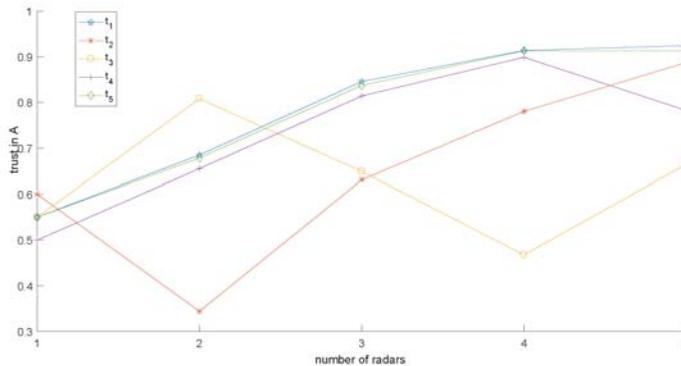


Figure 12. The trend of fusion results from different numbers of radars.

From Figure 12, we can say that the more radars that are in the identification system, the lower the uncertainty of the results are if all radars are working properly (as shown by the results of the moment t_1). Why does this happen? As mentioned in Section 2, Dempster’s combination rule, which is a part of our method, has a characteristic: mutual support evidence is reinforced. Therefore, the greater the number of radars support “A”, the greater the trust given to “A” in the fusion results. However, what will happen if some radars are not working properly? Dempster’s combination rule has another characteristic: pieces of conflict evidence weaken each other. Therefore, the uncertainty of the fusion results will rise if some radars are not working properly. For example, radar supports proposition “B”, so we get unreasonable results if we just use the first two radars. At moment t_3 , radar 3 and radar 4 are not working properly, while moment t_4 is radar 5 and moment t_4 is also radar 5. From Figure 12, when we join the radars that are not working properly, trust given to “A” declines. Logically speaking, we can get more information if there are more radars in the identification system, and then fusion results obtained are more reliable. If the number of radars is too small, it is likely that the number of radars that are not working properly is almost the same as those working properly, even if the number of radars not working properly is higher. In addition, in this situation, there will be a high degree of uncertainty or even fusion results that are completely wrong.

This is also the case for the number of sampled moments. The more points that are sampled in the time domain, the more information is obtained. Looking at the data from the last column of Table 7, if we only sample at one moment, the fusion results are likely to have higher uncertainty. Therefore, in general, whether all the radars are working properly or some radars are not working properly, we should try to use a larger number of radars and more moments to get more information. It is obvious that the greater amount of information, the more favorable it is for us to get reasonable results.

6. Conclusions

Specific Emitter Identification technology is one of the key technologies of the modern electronic warfare system and electromagnetic environment monitor. Most of the previous methods have been proposed from the point of view of feature signal processing. In addition, they did not consider dealing with uncertain information. Therefore, in this paper, we started from the point of view of information fusion and proposed a new identification method based on D–S evidence theory, as it is a powerful tool to deal with uncertain information. At first, we simulated the actual radar data with assumed BPAs. Secondly, we changed the number of radars and movement trajectory, respectively, to simulate the effect of the number and relative position on the final results. From the simulation results, final results of the proposed method have low uncertainty and good robustness. In addition, the greater the number of radars and sampling points is, the more information we obtained, making the fusion results more reliable. Thirdly, from the calculation principle of this proposed method, it considers not only the relationship between the BPAs (evidence), but also the performance of radars themselves to generate weights. This is more comprehensive than the previous methods, as most of the previous methods only considered one of the two influencing factors. Fourthly, since the distance between radar and target will change over time, we join time-domain evidence fusion. Time-domain fusion can prevent unreasonable results that generate occasionally at a single moment, which misleads the final decision-making.

In summary, vertical strategy is applied firstly, that is, the fusion results are generated by assigning weights at a single moment and fusing the modified average evidence. We then use the horizontal strategy (time-domain recursive fusion) to demonstrate the inheritance and updating of the fusion results. Finally, the computational burden of the proposed method mainly exists in Dempster's combination rule. In addition, the computational complexity of Dempster's combination rule increases exponentially with the cardinality of the discernment frame. In this paper, computational burden is small as we only use BPAs of five radars at five moments to simulate the practical situation in MatLab (MATLAB R2016b, MathWorks, Natick, MA, USA). In real conditions, hardware acceleration can be used to reduce the computational burden. This is the next step we want to study. In addition, the spatial positions of the five radars in the simulation of this paper are man-made. In future work, we can consider designing a new algorithm from the perspective of signal processing to realize the optimal configuration of multiple radars, so as to generate reasonable BPAs and then make the identification results more reliable.

Acknowledgments: The work is partially supported by the National Natural Science Foundation of China (Grant No. 61671384), the Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016JM6018), the Aviation Science Foundation (Program No. 20165553036), and the Fundamental Research Funds for the Central Universities (Program No. 3102017OQD020).

Author Contributions: W.J. and Y.C. proposed the method; W.J., L.Y., Y.C. and Z.H analyzed the results of simulation; Y.C. wrote the paper; W.J. and L.Y. revised and improved the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Du, P.; Liu, S.; Xia, J.; Zhao, Y. Information fusion techniques for change detection from multi-temporal remote sensing images. *Inf. Fusion* **2013**, *14*, 19–27.
- Dia, H.; Thomas, K. Development and evaluation of arterial incident detection models using fusion of simulated probe vehicle and loop detector data. *Inf. Fusion* **2011**, *12*, 20–27.
- Ehlenbröker, J.F.; Mönks, U.; Lohweg, V. Sensor defect detection in multisensor information fusion. *J. Sens. Sens. Syst.* **2016**, *5*, 337–353.
- Yang, G.; Lin, Y.; Bhattacharya, P. A driver fatigue recognition model based on information fusion and dynamic Bayesian network. *Inf. Sci.* **2010**, *180*, 1942–1954.
- Sun, S.; Tian, T.; Lin, H. Optimal Linear Estimators for Systems with Finite-Step Correlated Noises and Packet Dropout Compensations. *IEEE Trans. Signal Proc.* **2016**, *64*, 5672–5681.
- Xu, Z.Y.; Xu, W.S.; Shao, W.H.; Zeng, Z.Y. Real-time pricing control on generation-side: optimal demand-tracking model and information fusion estimation solver. *IEEE Trans. Power Syst.* **2014**, *29*, 1522–1535.
- Li, X.; Dick, A.; Shen, C.; Zhang, Z.; van den Hengel, A.; Wang, H. Visual tracking with spatio-temporal Dempster–Shafer information fusion. *IEEE Trans. Image Proc.* **2013**, *22*, 3028–3040.
- Chin, T.; Xiong, K.; Blasch, E. Nonlinear target tracking for threat detection using rssi and optical fusion. In Proceedings of the 2015 18th International Conference on Information Fusion (Fusion), Washington, DC, USA, 6–9 July 2015; pp. 1946–1953.
- Jin, X.; Du, J.; Bao, J. Maneuvering target tracking by adaptive statistics model. *J. China Univ. Posts Telecommun.* **2013**, *20*, 108–114.
- Yang, F.; Ji, L.e.; Liu, S.; Feng, P. A fast and high accuracy registration method for multi-source images. *Optik* **2015**, *126*, 3061–3065.
- Wu, F.; Wei, H.; Wang, X. Correction of image radial distortion based on division model. *Opt. Eng.* **2017**, *56*, 013108.
- Yang, F.; Wei, H. Fusion of infrared polarization and intensity images using support value transform and fuzzy combination rules. *Infrared Phys. Technol.* **2013**, *60*, 235–243.
- Xu, X.; Feng, H.; Wang, Z.; Wen, C. An information fusion method of fault diagnosis based on interval basic probability assignment. *Chin. J. Electron.* **2011**, *20*, 255–260.
- Jiang, W.; Xie, C.; Zhuang, M.; Shou, Y.; Tang, Y. Sensor data fusion with Z-numbers and its application in fault diagnosis. *Sensors* **2016**, *16*, 1509.
- Deng, X.; Jiang, W. An evidential axiomatic design approach for decision making using the evaluation of belief structure satisfaction to uncertain target values. *Int. J. Intell. Syst.* **2017**, Article in press, doi:10.1002/int.21929.
- Basir, O.; Yuan, X. Engine fault diagnosis based on multi-sensor information fusion using Dempster–Shafer evidence theory. *Inf. Fusion* **2007**, *8*, 379–386.
- Islam, M.S.; Sadiq, R.; Rodriguez, M.J.; Najjaran, H.; Hoorfar, M. Integrated decision support system for prognostic and diagnostic analyses of water distribution system failures. *Water Resour. Manag.* **2016**, *30*, 2831–2850.
- Ma, J.; Liu, W.; Miller, P.; Zhou, H. An evidential fusion approach for gender profiling. *Inf. Sci.* **2016**, *333*, 10–20.
- Jiang, W.; Wei, B.; Zhan, J.; Xie, C.; Zhou, D. A visibility graph power averaging aggregation operator: A methodology based on network analysis. *Comput. Ind. Eng.* **2016**, *101*, 260–268.
- Deng, X.; Jiang, W.; Zhang, J. Zero-sum matrix game with payoffs of Dempster–Shafer belief structures and its applications on sensors. *Sensors* **2017**, *17*, 922.
- Sun, S.; Wang, G. Modeling and estimation for networked systems with multiple random transmission delays and packet losses. *Syst. Control Lett.* **2014**, *73*, 6–16.
- Xu, S.; Jiang, W.; Deng, X.; Shou, Y. A modified Physarum-inspired model for the user equilibrium traffic assignment problem. *Appl. Math. Model.* **2017**, in press, doi: 10.1016/j.apm.2017.07.032.
- Dempster, A.P. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339.
- Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press Princeton: Princeton, NJ, USA, 1976; Volume 1.
- Leung, Y.; Ji, N.N.; Ma, J.H. An integrated information fusion approach based on the theory of evidence and group decision-making. *Inf. Fusion* **2013**, *14*, 410–422.

26. Zhang, X.; Deng, Y.; Chan, F.T.S.; Adamatzky, A.; Mahadevan, S. Supplier selection based on evidence theory and analytic network process. *Proc. Inst. Mech. Eng. Part B* **2016**, *230*, 562–573.
27. Jiang, W.; Wang, S. An Uncertainty Measure for Interval-valued Evidences. *Int. J. Comput. Comun.* **2017**, *12*, 631–644.
28. Denooux, T.; Sriboonchitta, S.; Kanjanatarakul, O. Evidential clustering of large dissimilarity data. *Knowl. Based Syst.* **2016**, *106*, 179–195.
29. Yager, R.R.; Alajlan, N. Evaluating belief structure satisfaction to uncertain target values. *IEEE Trans. Cybern.* **2016**, *46*, 869–877.
30. Deng, Y.; Shi, W.; Zhu, Z.; Liu, Q. Combining belief functions based on distance of evidence. *Decis. Support Syst.* **2004**, *38*, 489–493.
31. Dambreville, F.; Celeste, F.; Dezert, J.; Smarandache, F. Probabilistic PCR6 fusion rule. *Adv. Appl. DSMT Inf. Fusion* **2009**, *3*, 137–160.
32. Yager, R.R. On the Dempster-Shafer framework and new combination rules. *Inf. Sci.* **1987**, *41*, 93–137.
33. Dubois, D.; Prade, H. Representation and combination of uncertainty with belief functions and possibility measures. *Comput. Intell.* **1998**, *4*, 244–264.
34. Deng, Y. Deng entropy. *Chaos Soliton. Fract.* **2016**, *12*, 549–553.
35. Jiang, W.; Zhan, J. A modified combination rule in generalized evidence theory. *Appl. Intell.* **2017**, *46*, 630–640.
36. Wang, J.; Xiao, F.; Deng, X.; Fei, L.; Deng, Y. Weighted evidence combination based on distance of evidence and entropy function. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 3218784.
37. Fu, D.; Chen, B.; Zhang, H.; Wang, J.; Black, T.A.; Amiro, B.D.; Bohrer, G.; Bolstad, P.; Coulter, R.; Rahman, A.F.; et al. Estimating landscape net ecosystem exchange at high spatial-temporal resolution based on Landsat data, an improved upscaling model framework, and eddy covariance flux measurements. *Remote Sens. Environ.* **2014**, *141*, 90–104.
38. Zhang, W.; Li, A.; Jin, H.; Bian, J.; Zhang, Z.; Lei, G.; Qin, Z.; Huang, C. An enhanced spatial and temporal data fusion model for fusing Landsat and MODIS surface reflectance to generate high temporal Landsat-like data. *Remote Sens.* **2013**, *5*, 5346–5368.
39. Zhu, X.; Chen, J.; Gao, F.; Chen, X.; Masek, J.G. An enhanced spatial and temporal adaptive reflectance fusion model for complex heterogeneous regions. *Remote Sens. Environ.* **2010**, *114*, 2610–2623.
40. Hong, L.; Lynch, A. Recursive temporal-spatial information fusion with applications to target identification. *Aerosp. Electron. Syst. IEEE Trans.* **1993**, *29*, 435–445.
41. Hong, Z.Y. Research on temporal-spatial information fusion model based on D-S theory. *Signal Proc.* **2011**, *27*, 14–19.
42. Matuszewski, J. Specific emitter identification. In Proceedings of the 2008 International Radar Symposium, Rome, Italy, 26–30 May, 2008; pp. 1–4.
43. Dudczyk, J. Radar emission sources identification based on hierarchical agglomerative clustering for large data sets. *J. Sens.* **2016**, *2016*, 1879327.
44. Dudczyk, J. A method of feature selection in the aspect of specific identification of radar signals. *Bull. Pol. Acad. Sci. Tech. Sci.* **2017**, *65*, 113–119.
45. Dudczyk, J.; Kawalec, A.; Cyrek, J. Applying the distance and similarity functions to radar signals identification. In Proceedings of the 2008 International Radar Symposium, Wroclaw, Poland, 21–23 May 2008; pp. 1–4.
46. He, Z.; Jiang, W. Quantum mechanical approach to modelling reliability of sensor report. *IEEE Sens. Lett.* **2017**, *1*, 7500304.
47. Jiang, W. A correlation coefficient of belief functions. *arXiv* **2016**, arXiv:1612.05497.
48. Jiang, W.; Xie, C.; Zhuang, M.; Tang, Y. Failure Mode and Effects Analysis based on a novel fuzzy evidential method. *Appl. Soft Comput.* **2017**, *57*, 672–683.
49. Fu, C.; Yang, J.; Yang, S. A group evidential reasoning approach based on expert reliability. *Eur. J. Oper. Res.* **2015**, *246*, 886–893.
50. Deng, X.; Xiao, F.; Deng, Y. An improved distance-based total uncertainty measure in belief function theory. *Appl. Intell.* **2017**, *46*, 898–915.
51. Zadeh, L.A. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. *Ai Mag.* **1986**, *7*, 85–90.

52. Jousselme, A.L.; Grenier, D.; Bossé, É. A new distance between two bodies of evidence. *Inf. Fusion* **2001**, *2*, 91–101.
53. Liu, W. Analyzing the degree of conflict among belief functions. *Artif. Intell.* **2006**, *170*, 909–924.
54. Jiang, W.; Wang, S.; Liu, X.; Zheng, H.; Wei, B. Evidence conflict measure based on OWA operator in open world. *PLoS ONE* **2017**, *12*, e0177828.
55. Mo, H.; Lu, X.; Deng, Y. A generalized evidence distance. *J. Syst. Eng. Electron.* **2016**, *27*, 470–476.
56. Martin, A., About conflict in the theory of belief functions. In *Belief Functions: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 161–168.
57. Perez, A.; Tabia, H.; Declercq, D.; Zanotti, A. Using the conflict in Dempster–Shafer evidence theory as a rejection criterion in classifier output combination for 3D human action recognition. *Image Vision Comput.* **2016**, *55*, 149–157.
58. Bolotin, A. Quantum mechanical approach to fuzzy logic modelling. *Math. Comput. Model.* **2001**, *34*, 937–945.
59. Watson, G.N. *A Treatise on the Theory of Bessel Functions*; Cambridge University Press: Cambridge, UK, 1995.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Fuzzy Risk Evaluation in Failure Mode and Effects Analysis Using a D Numbers Based Multi-Sensor Information Fusion Method

Xinyang Deng * and Wen Jiang *

School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China

* Correspondence: xinyang.deng@nwpu.edu.cn (X.D.); jiangwen@nwpu.edu.cn (W.J.);

Tel.: +86-29-8843-1267 (W.J.)

Received: 30 August 2017; Accepted: 11 September 2017; Published: 12 September 2017

Abstract: Failure mode and effect analysis (FMEA) is a useful tool to define, identify, and eliminate potential failures or errors so as to improve the reliability of systems, designs, and products. Risk evaluation is an important issue in FMEA to determine the risk priorities of failure modes. There are some shortcomings in the traditional risk priority number (RPN) approach for risk evaluation in FMEA, and fuzzy risk evaluation has become an important research direction that attracts increasing attention. In this paper, the fuzzy risk evaluation in FMEA is studied from a perspective of multi-sensor information fusion. By considering the non-exclusiveness between the evaluations of fuzzy linguistic variables to failure modes, a novel model called D numbers is used to model the non-exclusive fuzzy evaluations. A D numbers based multi-sensor information fusion method is proposed to establish a new model for fuzzy risk evaluation in FMEA. An illustrative example is provided and examined using the proposed model and other existing method to show the effectiveness of the proposed model.

Keywords: fuzzy risk evaluation; failure mode and effects analysis; multi-sensor information fusion; D numbers; dempster-shafer evidence theory; fuzzy uncertainty

1. Introduction

Failure mode and effects analysis (FMEA) is a widely used technology in many fields to identify potential failures or errors and further improve the reliability of systems by avoiding the occurrence of these failures or errors [1–3]. Risk evaluation is a crucial step in FMEA, which aims to identify failure modes with high risk so as to perfect system design to eliminating the risk [4,5]. In FMEA, risk priority number (RPN) approach is a classical method for the risk evaluation [6,7]. Since having clear physical meaning and easy to implement, the RPN approach has been received extensively concern and application. However, there still are some shortcomings in the RPN approach [8,9], for example the possible missing of risk factors, without considering the relative importance of risk factors, and so on. Among these drawbacks, failing to address the uncertainty in risk evaluation is one of the most concerned, and has attracted increasing attention [10–13].

In the risk evaluation of FMEA, domain experts' knowledge and evaluations play a very importance role. Because there are human being's judgments, it inevitably involves various types of uncertainties such as ignorance and fuzziness. Fuzzy set theory [14] provides a useful framework to describe the uncertain information. Therefore, risk evaluation under fuzzy environment, also known as fuzzy risk evaluation, has become an important research direction in FMEA. Many technologies have been developed to solve the problem of fuzzy risk evaluation in FMEA. Chin et al. [15] presented a data envelopment analysis (DEA) based FMEA to determine the risk priorities of failure modes. Jee et al. [16] proposed a fuzzy inference system (FIS)-based RPN model for the prioritization of failures.

In [17], the authors have given applied a model of evolving tree to allow the failure modes in FMEA to be clustered and visualized. Reference [18] gives a detained literature review to the risk evaluation approaches in FMEA. By summarizing the existing approaches [10,18], one of the main branches is to regard the fuzzy risk evaluation in FMEA as a multiple criteria decision making (MCDM) problem under fuzzy environment. Many MCDM or multi-sensor information fusion technologies [19–26] have been used in FMEA, such as TOPSIS [27], VIKOR [28], evidential reasoning [29], and so on.

In this paper, we address the fuzzy risk evaluation in FMEA from a perspective of multi-sensor information fusion. Each risk factor in FMEA is regarded as a sensor or information source that yields an evaluation regarding the risk of each failure mode. Then, the risk evaluation of every failure mode therefore becomes the process of fusing these evaluations generated from the information sources that correspond to risk factors. Different from existing multi-sensor information fusion method used in FMEA, in this paper the non-exclusiveness between the evaluations of fuzzy linguistic variables to failure modes is taken into consideration. A novel model called D numbers [30–33] which is an extension of Dempster-Shafer evidence theory [34,35] is used to model the non-exclusive fuzzy evaluations. At first, a new D numbers based multi-sensor information fusion method is proposed. Then, the proposed multi-sensor information fusion method is applied to FMEA, which results in a novel model for fuzzy risk evaluation. At last, an illustrative example is given to demonstrate the effectiveness of the proposed model.

The rest of the paper is organized as follows. Section 2 gives a brief introduction about fuzzy set theory, RPN approach, as well as Dempster-Shafer evidence theory and D numbers. In Section 3, a novel multi-sensor information fusion method is proposed based on D numbers. Then, the new model for fuzzy risk evaluation in FMEA is presented in Section 4. Section 5 gives an illustrative example of the proposed model to show its effectiveness. Lastly, Section 6 concludes this paper. In addition, the notations of this paper are briefly introduced here: \tilde{A} represents a fuzzy set or fuzzy number, and $\mu_{\tilde{A}}$ is its corresponding membership function, and $Area_{\tilde{A}}$ represents the area of \tilde{A} in the graph; Ω and Θ stand for the frame of discernment in Dempster-Shafer theory and D numbers respectively, m represents a mass function and D is a D number; $u_{-E}(\tilde{A}, \tilde{B})$ is the non-exclusive degree between \tilde{A} and \tilde{B} ; $BetP$ represents a distribution of pignistic probabilities; $C_{\tilde{F}}$ is the defuzzified value of fuzzy number \tilde{F} .

2. Preliminaries

2.1. Fuzzy Set Theory

Fuzzy set theory was first introduced by Zadeh [14] in 1965 to deal with the uncertainty information. In some real application environments, the states are subjective concepts which are too complex or too ill-defined to be reasonably described in conventional quantitative expressions. In those situation, fuzzy set theory provides an efficiently simple way to express the vagueness or imprecise information [36,37].

Definition 1. Let X be the universe of discourse, a fuzzy set \tilde{A} is characterized by a membership function $\mu_{\tilde{A}}$ satisfying

$$\mu_{\tilde{A}} : X \rightarrow [0, 1] \tag{1}$$

where $\mu_{\tilde{A}}(x)$ is called the membership degree of $x \in X$ belonging to fuzzy set \tilde{A} .

For a finite set $A = \{x_1, \dots, x_i, \dots, x_n\}$, the fuzzy set $(\tilde{A}, \mu_{\tilde{A}})$ is often denoted by $\left\{ \mu_{\tilde{A}}(x_1)/_{x_1}, \dots, \mu_{\tilde{A}}(x_i)/_{x_i}, \dots, \mu_{\tilde{A}}(x_n)/_{x_n} \right\}$. It is easily found that a fuzzy set is described entirely by its membership function. When $\mu_{\tilde{A}}$ takes value from $\{0, 1\}$, fuzzy set \tilde{A} degenerates into a classical set. A fuzzy number \tilde{A} is a fuzzy subset of the real number R , and its membership function is

$$\mu_{\tilde{A}}(x) : R \rightarrow [0, 1] \tag{2}$$

where x is a real number and there definitely exists an element x_0 such that $\mu_{\tilde{A}}(x_0) = 1$. Triangular and trapezoidal fuzzy numbers are the most widely used fuzzy numbers, and the former can be regarded as the special case of the latter. A trapezoidal fuzzy number is usually denoted as $\tilde{A} = (a_1, a_2, a_3, a_4)$, as graphically shown in Figure 1, which has the following membership function

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x \leq a_1; \\ \frac{x-a_1}{a_2-a_1} & a_1 \leq x \leq a_2; \\ 1, & a_2 \leq x \leq a_3; \\ \frac{a_4-x}{a_4-a_3}, & a_3 \leq x \leq a_4; \\ 0, & a_4 \leq x. \end{cases} \quad (3)$$

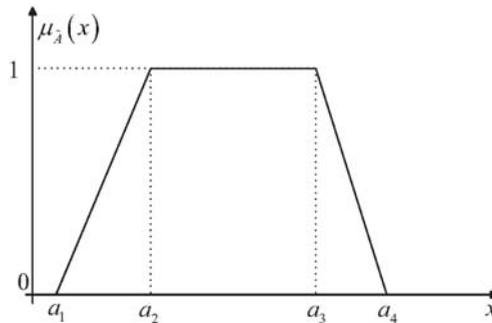


Figure 1. Graphically presentation of the trapezoidal fuzzy number.

In theory and practice, fuzzy numbers are usually associated with linguistic variables to express the fuzzy evaluation to objects. A linguistic variable is a variable whose values are represented by words or sentences in a natural or artificial language, for example “Very Low”, “Low”, “Medium”, “High”, “Very High”, where there values are usually expressed by fuzzy numbers.

2.2. Risk Priority Number Approach in FMEA

Risk priority number (RPN) approach is a traditional risk evaluation method in FMEA. In this approach, the risk priority of each failure mode is expressed by a RPN which is defined by the following formula

$$RPN = O \times S \times D \quad (4)$$

where O is the probability of occurrence of a failure mode, S is the severity of the failure effect, and D is the probability of a failure mode not being detected before it occurs. In the RPN approach, each factor among O , S , and D is evaluated by 10 rankings, as shown in Tables 1–3. The larger the RPN value, the higher the risk priority of a failure mode.

Table 1. Assessment rankings for occurrence in FMEA [38,39].

Ranking	Probability of Occurrence	Possible Failure Rate
10	Extremely high: failure almost inevitable	$\geq 1/2$
9	Very high	1/3
8	Repeated failures	1/8
7	High	1/20
6	Moderately high	1/80
5	Moderate	1/400
4	Relatively low	1/2000
3	Low	1/15,000
2	Remote	1/150,000
1	Nearly impossible	$\leq 1/1,500,000$

Table 2. Assessment rankings for severity in FMEA [38,39].

Ranking	Effect	Severity of Effect
10	Hazardous without warning	Very high severity ranking when a potential failure mode affects safe vehicle operation and/or involves noncompliance with government regulations without warning
9	Hazardous with warning	Very high severity ranking when a potential failure mode affects safe vehicle operation and/or involves noncompliance with government regulations with warning
8	Very high	Vehicle/item inoperable, with loss of primary function
7	High	Vehicle/item operable, but at reduced level of performance. Customer dissatisfied
6	Moderate	Vehicle/item operable, but comfort/convenience item(s) inoperable. Customer experiences discomfort
5	Low	Vehicle/item operable, but comfort/convenience item(s) operable at reduced level of performance. Customer experiences some dissatisfaction.
4	Very low	Cosmetic defect in finish, fit and finish/squeak or rattle item that does not conform to specifications. Defect noticed by most customers
3	Minor	Cosmetic defect in finish, fit and finish/squeak or rattle item that does not conform to specifications. Defect noticed by average customer
2	Very minor	Cosmetic defect in finish, fit and finish/squeak or rattle item that does not conform to specifications. Defect noticed by discriminating customers
1	None	No effect

Table 3. Assessment rankings for detection in FMEA [38,39].

Ranking	Detection	Criteria
10	Absolutely impossible	Design control will not and/or cannot detect a potential cause/mechanism and subsequent failure mode; or there is no design control
9	Very remote	Very remote chance the design control will detect a potential cause/mechanism and subsequent failure mode
8	Remote	Remote chance the design control will detect a potential cause/mechanism and subsequent failure mode
7	Very low	Very low chance the design control will detect a potential cause/mechanism and subsequent failure mode
6	Low	Low chance the design control will detect a potential cause/mechanism and subsequent failure mode
5	Moderate	Moderate chance the design control will detect a potential cause/mechanism and subsequent failure mode
4	Moderately high	Moderately high chance the design control will detect a potential cause/mechanism and subsequent failure mode
3	High	High chance the design control will detect a potential cause/mechanism and subsequent failure mode
2	Very high	Very high chance the design control will detect a potential cause/mechanism and subsequent failure mode
1	Almost certain	Design control will almost certainly detect a potential cause/mechanism and subsequent failure mode

Although the RPN approach is easy to use, but it still has some shortcomings that are criticized in many studies [8,9]. For example, three risk factors O, S, D are considered with equal importance;

Traditional RPN approach only considers three factors but ignores other possible influential factors to different application environment; Various sets of O, S and D may produce an identical RPN value, but their hidden risk implications may be different. Please refer to [8,9] for more details on the drawbacks of RPN approach.

2.3. Dempster-Shafer Evidence Theory and D Numbers

Dempster-Shafer evidence theory [34,35], also called Dempster-Shafer (D-S) theory or evidence theory, is a popular theory to deal with uncertain information [40–45]. Compared with traditional probability theory, this theory has an advantage of directly expressing the “uncertainty” by assigning the probability to the set composed of multiple objects, it therefore has attracted increasing interest in uncertainty reasoning and modelling [46–52] and been further extended such as generalized evidence theory (GET) in open world [53–55].

Let Ω be a set of mutually exclusive and collectively exhaustive events, indicated by $\Omega = \{E_1, E_2, \dots, E_i, \dots, E_N\}$, where set Ω is called a frame of discernment (FOD). The power set of Ω is indicated by 2^Ω , namely $2^\Omega = \{\emptyset, \{E_1\}, \dots, \{E_N\}, \{E_1, E_2\}, \dots, \{E_1, E_2, \dots, E_i\}, \dots, \Omega\}$. In D-S theory, the uncertain information is modelled by mass functions.

Definition 2. Given a FOD Ω , a mass function is a mapping $m : 2^\Omega \rightarrow [0, 1]$, such that

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \in 2^\Omega} m(A) = 1 \tag{5}$$

where set A with $m(A) > 0$ is called a focal element, and the assigned $m(A)$ measures the belief exactly assigned to A and represents how strongly the evidence supports A . A mass function is also called a basic probability assignment (BPA).

Considering two pieces of evidence indicated by m_1 and m_2 , Dempster’s rule of combination can be used to combine them. This rule assumes that these mass functions are mutually independent. The Dempster’s rule of combination, denoted by $m = m_1 \oplus m_2$, is defined as follows:

$$m(A) = \begin{cases} \frac{1}{1-K} \sum_{B \cap C = A} m_1(B)m_2(C), & A \neq \emptyset \\ 0, & A = \emptyset \end{cases} \tag{6}$$

with

$$K = \sum_{B \cap C = \emptyset} m_1(B)m_2(C). \tag{7}$$

In order to make decision in terms of a mass function [56], an approach, called pignistic probability transformation (PPT), is proposed by Smets and Kennes [57] to derive a probability distribution from a mass function. Let m be a mass function or BPA on FOD Ω , a PPT function $BetP_m : \Omega \rightarrow [0, 1]$ associated to m is defined by

$$BetP_m(x) = \sum_{x \in A, A \subseteq \Omega} \frac{1}{|A|} \frac{m(A)}{1 - m(\emptyset)}, \tag{8}$$

where $m(\emptyset) \neq 1$ and $|A|$ is the cardinality of proposition A .

Although D-S theory provides a good framework for uncertainty reasoning, this theory is also constrained by many strong hypotheses and hard constraints which limit its development and application to a large extent. For one hand, the elements in the FOD are required to be mutually exclusive. It is called exclusiveness hypothesis. For another, the sum of basic probabilities of a mass function must be equal to 1, which is called completeness constraint. To overcome these existing shortcomings in D-S theory and enhance its capability in expressing uncertain information, a novel

model, named as D numbers, has been proposed recently [30–33,58]. D numbers relax mass function’s exclusiveness hypothesis and BPA’s completeness constraint.

Definition 3. Let $\Theta = \{F_1, F_2, \dots, F_N\}$ be a nonempty set satisfying $F_i \neq F_j$ if $i \neq j, \forall i, j = \{1, \dots, N\}$, a D number is a mapping formulated by

$$D : 2^\Theta \rightarrow [0, 1] \tag{9}$$

with

$$\sum_{B \subseteq \Theta} D(B) \leq 1 \quad \text{and} \quad D(\emptyset) = 0 \tag{10}$$

where \emptyset is the empty set and B is a subset of Θ .

If $\sum_{B \subseteq \Theta} D(B) = 1$, the information is said to be complete; if $\sum_{B \subseteq \Theta} D(B) < 1$, the information is said to be incomplete. If a D number is of complete information, it means that the D number is generated from an environment with the close-world assumption. By contrast, a D number with incomplete information is corresponding to the open-world assumption. With respect to the open-world assumption, Smets [57] proposed a transferable belief model (TBM) which allows $m(\emptyset) > 0$. Compared with the TBM, in D numbers the open-world environment is implemented by letting $\sum_{B \subseteq \Theta} D(B) < 1$. What’s more important, in the TBM each mass function is defined on a FOD which requires the internal elements are mutually exclusive. However, in D numbers the exclusiveness hypothesis is removed, i.e., the elements in Θ do not require mutual exclusiveness for D numbers.

3. Proposed Multi-Sensor Information Fusion Method Based on D Numbers

Let us consider a multiple criteria decision making (MCDM) problem, where each criterion can be regarded as an independent sensor or information source. Therefore, the process of resolving the MCDM problem can be treated as a process of multi-sensor information fusion. Assume there are p alternatives, indicated by $A_i, i = 1, \dots, p$, and q criteria, denoted as $C_j, j = 1, \dots, q$, and the weight of each criterion is $w_j, j = 1, \dots, q$. Due to the uncertainty of decision-making environment, the evaluation to alternative A_i on criterion C_j is expressed as a D number indicated by D_{ij} , thus the decision matrix is represented as

$$M = \begin{matrix} & & C_1 & \cdots & C_q \\ \begin{matrix} A_1 \\ \vdots \\ A_p \end{matrix} & \left(\begin{matrix} D_{11} & \cdots & D_{1q} \\ \cdots & \cdots & \cdots \\ D_{p1} & \cdots & D_{pq} \end{matrix} \right) \end{matrix} \tag{11}$$

In this paper, we assume that each evaluation D_{ij} in the decision matrix M is information-complete, i.e., $\sum_k D_{ij}(B_k) = 1$ for any $i = 1, \dots, p$ and $j = 1, \dots, q$. Now the overall objective is to find the best alternative according to the decision matrix M and criteria’s weights mentioned above. In this study, we develop a new multi-sensor information fusion method based on D numbers to solve that problem. The key points of the proposed approach are presented as follows.

3.1. Non-Exclusiveness in D Numbers

Since the evaluations are in the form of D numbers and the theory of D numbers is found on the basis of non-exclusiveness assumption, the first step is to calculate the non-exclusive degrees in D numbers. The non-exclusiveness is the opposite of exclusiveness, representing a potential connection between elements in D numbers framework. By contrast, the exclusiveness refers to the characteristic that one object excludes the others, which is an either-or related thing but not the similarity.

Definition 4. Let B_i and B_j be two non-empty elements belonging to 2^Θ , the non-exclusive degree between B_i and B_j is characterized by a fuzzy membership function u_{-E} :

$$u_{-E} : 2^\Theta \times 2^\Theta \rightarrow [0, 1] \tag{12}$$

with

$$u_{-E}(B_i, B_j) = \begin{cases} 1, & B_i \cap B_j \neq \emptyset \\ p, & p \in [0, 1], B_i \cap B_j = \emptyset \end{cases} \tag{13}$$

and

$$u_{-E}(B_i, B_j) = u_{-E}(B_j, B_i). \tag{14}$$

If letting the exclusive degree between B_i and B_j be denoted as u_E , then $u_E = 1 - u_{-E}$.

In our previous study [59], a simple approach was proposed to determine the non-exclusive degrees in D numbers. In that approach it assumes that all non-exclusive degrees among elements in FOD Θ have already been determined, then each exclusive degree in the power set space 2^Θ can be calculated by the following formula:

$$u_{-E}(B_i, B_j) = \max_{x \in B_i, y \in B_j} \{u_{-E}(x, y)\}, B_i, B_j \in 2^\Theta. \tag{15}$$

An illustrative example regarding the calculation of non-exclusive degrees is given as follows.

Example 1. Supposing each evaluation in decision matrix M shown in Equation (11) is defined on a set of linguistic variables $\Theta = \{VL, L, ML, M, MH, H, VH\}$ in which every linguistic variable is represented by a trapezoidal fuzzy number given in Table 4 and graphically presented as Figure 2.

Table 4. Linguistic variables for the evaluation.

Linguistic Variables	Fuzzy Numbers
Very Low (VL)	(0, 0, 1, 2)
Low (L)	(1, 2, 2, 3)
Medium Low (ML)	(2, 3, 4, 5)
Medium (M)	(4, 5, 5, 6)
Medium High (MH)	(5, 6, 7, 8)
High (H)	(7, 8, 8, 9)
Very High (VH)	(8, 9, 10, 10)

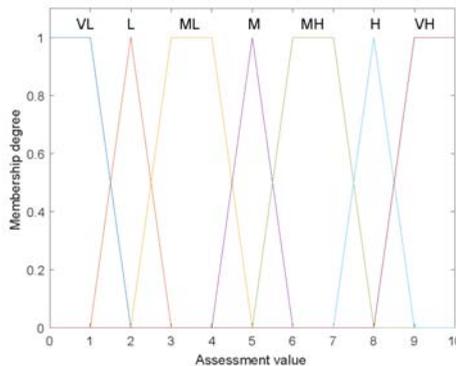


Figure 2. Graphically presentation of fuzzy linguistic variables in Table 4.

The set Θ is seen as the FOD. At first, let us calculate the non-exclusive degrees between elements in FOD Θ . In this paper, an approach based on fuzzy numbers' areas is utilized. Assume the areas of fuzzy numbers \tilde{A} and \tilde{B} are respectively denoted as $Area_{\tilde{A}}$ and $Area_{\tilde{B}}$, and the area of the overlap of \tilde{A} and \tilde{B} is $Area_{\tilde{A} \cap \tilde{B}}$, then the non-exclusive degree between \tilde{A} and \tilde{B} is defined as

$$u_{-E}(\tilde{A}, \tilde{B}) = \frac{Area_{\tilde{A} \cap \tilde{B}}}{Area_{\tilde{A}} + Area_{\tilde{B}} - Area_{\tilde{A} \cap \tilde{B}}}. \tag{16}$$

According to Equation (16), each non-exclusive degree between elements in FOD Θ therefore can be obtained as shown in the following matrix

$$\begin{matrix}
 & VL & L & ML & M & MH & H & VH \\
 VL & \left(\begin{matrix} 1 & 0.1111 & 0 & 0 & 0 & 0 & 0 \\ 0.1111 & 1 & 0.0909 & 0 & 0 & 0 & 0 \\ 0 & 0.0909 & 1 & 0.0909 & 0 & 0 & 0 \\ 0 & 0 & 0.0909 & 1 & 0.0909 & 0 & 0 \\ 0 & 0 & 0 & 0.0909 & 1 & 0.0909 & 0 \\ 0 & 0 & 0 & 0 & 0.0909 & 1 & 0.1111 \\ 0 & 0 & 0 & 0 & 0 & 0.1111 & 1 \end{matrix} \right) \\
 L & & & & & & & \\
 ML & & & & & & & \\
 M & & & & & & & \\
 MH & & & & & & & \\
 H & & & & & & & \\
 VH & & & & & & &
 \end{matrix}$$

Having the above non-exclusive degree matrix of between elements in Θ , according to Equation (15), we can easily calculate the non-exclusive degree of any pair of elements in 2^Θ . For example, as for $\{L\}$ and $\{ML, M\}$, we have

$$\begin{aligned}
 u_{-E}(\{L\}, \{ML, M\}) &= \max \{u_{-E}(\{L\}, \{ML\}), u_{-E}(\{L\}, \{M\})\} \\
 &= \max \{0.0909, 0\} \\
 &= 0.0909.
 \end{aligned}$$

3.2. Fusing the Evaluations to the Same Alternative on Different Criteria

In order to implement the overall assessment to each alternative, all evaluations belonging to the same alternative on different criteria should be combined according to the perspective of multi-sensor information fusion. In this paper since the evaluations are given in the form of D numbers, it becomes the fusion of D numbers. In our recent study [59], a D numbers combination rule (DNCR) has been proposed from a perspective of conflict redistribution. The proposed DNCR is shown as follows.

Definition 5. Let D_1 and D_2 be two D numbers defined on Θ with $\sum_{B \subseteq \Theta} D_1(B) = 1$ and $\sum_{C \subseteq \Theta} D_2(C) = 1$, the combination of D_1 and D_2 , indicated by $D = D_1 \odot D_2$, is defined by

$$D(A) = \begin{cases} 0, & A = \emptyset \\ \frac{1}{1-K_D} \left(\sum_{B \cap C = A} u_{-E}(B, C) D_1(B) D_2(C) + \sum_{\substack{B \cup C = A \\ B \cap C = \emptyset}} u_{-E}(B, C) D_1(B) D_2(C) \right), & A \neq \emptyset \end{cases} \tag{17}$$

with

$$K_D = \sum_{B \cap C = \emptyset} (1 - u_{-E}(B, C)) D_1(B) D_2(C). \tag{18}$$

The above rule for D numbers is a generalization of Dempster's rule for the model of D numbers, because it can totally reduce to the classical Dempster's rule when $u_{-E}(B, C) = 0$ for any $B \cap C = \emptyset$. Different from the D-S theory, in this rule the impact of non-exclusiveness in D numbers is taken into consideration.

Although the rule defined in Definition 5 provides a solution for the combination of D numbers, it must point out that such rule does not preserve the associative property, i.e., $(D_1 \odot D_2) \odot D_3 \neq D_1 \odot (D_2 \odot D_3) \neq (D_1 \odot D_3) \odot D_2$, and it is only suitable for the combination of two D numbers. In order to implement the effective combination of multiple D numbers, a novel combination rule for multiple D numbers is developed in this paper by utilizing the idea of induced ordered weighted averaging (IOWA) operator [60] which imports an order variable compared with other aggregation operators [61,62].

Definition 6. Let D_1, D_2, \dots, D_n be n D numbers, and v_j be an order variable for each $D_j, j = 1, \dots, n$, therefore each piece of information is indicated by tuple $\langle v_j, D_{v_j} \rangle$. The combination operation of these information represented by D numbers is defined as a mapping Agg_D , such that

$$Agg_D(D_1, D_2, \dots, D_n) = [\dots [D_{\lambda_1} \odot D_{\lambda_2}] \odot \dots \odot D_{\lambda_n}] \tag{19}$$

where D_{λ_i} is the corresponding D_{v_j} in tuple $\langle v_j, D_{v_j} \rangle$ having the i -th largest order variable v_j .

In this paper, for the MCDM problem the weight of each criterion is regarded as the order variable of corresponding D numbers so as to fuse the evaluations to each alternative on multiple criteria. For each alternative $A_i, i = 1, \dots, p$, the obtained aggregated evaluation is denoted as D_i which is defined over the FOD Θ consisting of fuzzy linguistic variables.

3.3. Decision-Making Based on the Aggregated Evaluations under Fuzzy Environment

In this paper, each aggregated evaluation is also a D number, indicated by $D_i, i = 1, \dots, p$, which is defined on FOD Θ composed by fuzzy linguistic variables. Assume $\Theta = \{\theta_t, t = 1, \dots, l\}$, and each element θ_t is represented by a trapezoidal fuzzy number $\theta_t = (a_{t1}, a_{t2}, a_{t3}, a_{t4})$. Each D_i is firstly transformed to a distribution of pignistic probabilities, denoted as $BetP_i$, by means of the PPT as follows

$$BetP_i(\theta_t) = \sum_{\theta_t \in B, B \subseteq \Theta} \frac{D_i(B)}{|B|}, \quad t = 1, \dots, l. \tag{20}$$

Once the $BetP_i$ is obtained, it then be transformed to a fuzzy aggregated evaluation \tilde{F}_i to express the overall assessment to alternative i , represented as

$$\tilde{F}_i = (f_{i1}, f_{i2}, f_{i3}, f_{i4}), \quad i = 1, \dots, q \tag{21}$$

in which

$$f_{ik} = \sum_{t=1}^l BetP_i(\theta_t) \times a_{tk}, \quad k = 1, 2, 3, 4. \tag{22}$$

At last, these fuzzy aggregated evaluation $\tilde{F}_i, i = 1, \dots, p$, are converted to crisp values through a defuzzification process in order to rank all alternatives. Among existing defuzzification techniques, the centroid defuzzification approach is a common used one. Given a fuzzy number \tilde{F} with membership function $\mu_{\tilde{F}}(x)$, in terms of the centroid defuzzification approach we can have

$$C_{\tilde{F}} = \frac{\int x \mu_{\tilde{F}}(x) dx}{\int \mu_{\tilde{F}}(x) dx}, \tag{23}$$

where $C_{\tilde{F}}$ is the defuzzified value of \tilde{F} . In terms of the study in [63], while \tilde{F} is a trapezoidal fuzzy number indicated by (f_1, f_2, f_3, f_4) the centroid-based defuzzified value turns out to be

$$C_{\tilde{F}} = \frac{1}{3} \left(f_1 + f_2 + f_3 + f_4 - \frac{f_3 f_4 - f_1 f_2}{(f_3 + f_4) - (f_1 + f_2)} \right). \tag{24}$$

Via the defuzzification process, for each fuzzy aggregated evaluation \tilde{F}_i , a defuzzified value $C_{\tilde{F}_i}$ can be derived. The best alternative is finally determined by finding the one with the largest defuzzified value.

4. The Proposed Model for Fuzzy Risk Evaluation in FMEA

In terms of the multi-sensor information fusion method based on D numbers as presented above, a novel model for fuzzy risk evaluation in FMEA is proposed in this section. The flowchart of the proposed model is shown in Figure 3, which briefly contains four phases, namely identification, evaluation, preprocessing and ranking. The final output is the risk ranking of potential failure modes.

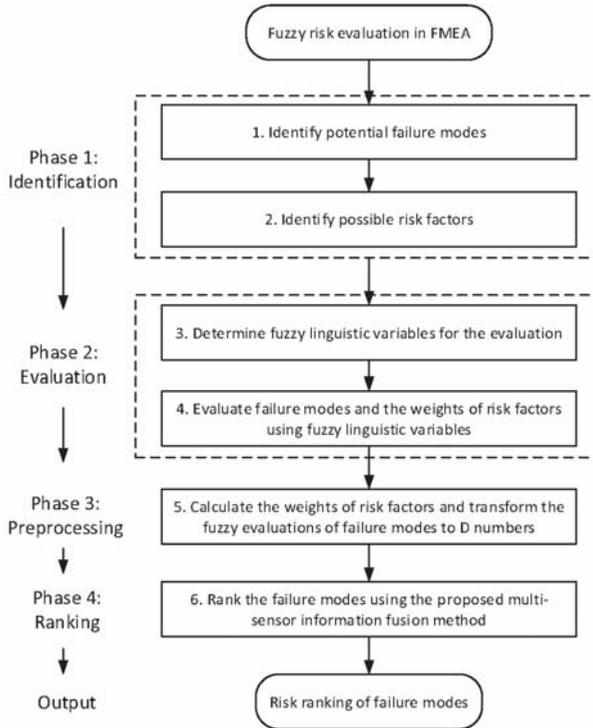


Figure 3. Flowchart of the proposed model for fuzzy risk evaluation in FMEA.

- Step 1: Identify all potential failure modes according to the practical FMEA environment.
- Step 2: Identify all possible risk factors for the risk evaluation task. In the RPN approach, the risk factors are probability of occurrence (O), severity of failure effect (S), and probability of a failure mode not being detected (D).
- Step 3: Determine fuzzy linguistic variables for the evaluation including evaluating failure modes and evaluating the weights of risk factors.
- Step 4: Evaluate failure modes and the weights of risk factors using fuzzy linguistic variables determined in the above step. These are usually given by domain experts of FMEA risk evaluation.
- Step 5: Calculate the weight of each risk factor and transform the fuzzy evaluations of failure modes on different risk factors to D numbers. This is a process of data preprocessing.
- Step 6: Rank the failure modes using the proposed multi-sensor information fusion method in above section so as to generate the risk ranking of all failure modes.

5. Illustrative Example

In the section, an illustrative example is given to show the effectiveness of the proposed model for fuzzy risk evaluation in FMEA. This example is original from literature [28]. In [28], the authors developed an extended VIKOR method for risk evaluation in FMEA under fuzzy environment. In this paper, we will solve the problem by using our proposed model and compare the obtained result with that of literature [28].

Step 1: Identify all potential failure modes. By following literature [28], a hospital wants to rank the most serious failure modes during general anesthesia process, and six potential failure modes are identified which are denoted as FM 1, FM 2, FM 3, FM 4, FM 5, FM 6.

Step 2: Identify all possible risk factors. In this application, the risk factors are consistent with the RPN approach, therefore there are three risk factors, namely O, S, D.

Step 3: Determine fuzzy linguistic variables for the evaluation. As for the evaluation of failure modes, a set of linguistic variables including Very Low (VL), Low (L), Medium Low (ML), Medium (M), Medium High (MH), Very High (VH) is used as shown in Table 4. In addition, for the evaluation of risk factors' weights, the fuzzy linguistic variables are given in Table 5.

Table 5. Linguistic variables for evaluating the weights of risk factors.

Linguistic Variables	Fuzzy Numbers
Very Low (VL)	(0, 0, 0.1, 0.2)
Low (L)	(0.1, 0.2, 0.2, 0.3)
Medium Low (ML)	(0.2, 0.3, 0.4, 0.5)
Medium (M)	(0.4, 0.5, 0.5, 0.6)
Medium High (MH)	(0.5, 0.6, 0.7, 0.8)
High (H)	(0.7, 0.8, 0.8, 0.9)
Very High (VH)	(0.8, 0.9, 1, 1)

Step 4: Evaluate failure modes and the weights of risk factors using fuzzy linguistic variables. As given in literature [28], a FMEA team of five decision makers, DM 1, DM 2, DM 3, DM 4, DM 5, is employed to evaluate failure modes and the weights of risk factors. With respect to risk factors' weights, all five decision makers' evaluations are given in Table 6. For the six failure modes, the evaluations from the FMEA team are given in Table 7.

Table 6. The evaluations to the weights of risk factors from the FMEA team.

Risk Factor	FMEA Team Member				
	DM 1	DM 2	DM 3	DM 4	DM 5
O	H	H	VH	H	MH
S	VH	VH	H	VH	VH
D	MH	MH	M	H	MH

Table 7. The evaluations to failure modes from the FMEA team.

	FM 1	FM 2	FM 3	FM 4	FM 5	FM 6
O						
DM 1	M	H	VH	M	M	MH
DM 2	M	MH	MH	M	ML	H
DM 3	M	H	VH	L	M	M
DM 4	MH	MH	VH	M	M	MH
DM 5	M	MH	VH	M	M	M

Table 7. Cont.

	FM 1	FM 2	FM 3	FM 4	FM 5	FM 6
S						
DM 1	ML	H	MH	M	M	H
DM 2	ML	MH	MH	M	MH	H
DM 3	ML	H	MH	ML	MH	H
DM 4	M	H	MH	M	M	H
DM 5	M	H	MH	M	M	H
D						
DM 1	M	M	MH	VL	L	L
DM 2	ML	M	M	ML	ML	M
DM 3	ML	ML	MH	VL	L	L
DM 4	ML	M	MH	ML	L	L
DM 5	ML	M	M	VL	L	VL

Step 5: At this step, the weight of each risk factor is calculated at first. Since the calculation of risk factors' weights is not the core concern of this paper, we simply continue to use the weights obtained in literature [28]. The importance of O is 0.768, and S 0.878, and D 0.650, therefore the weights of these risk factors are $w_O = 0.768 / (0.768 + 0.878 + 0.650) = 0.3345$, $w_S = 0.878 / (0.768 + 0.878 + 0.650) = 0.3824$, $w_D = 0.650 / (0.768 + 0.878 + 0.650) = 0.2831$. Secondly, let us transform the fuzzy evaluations of failure modes on risk factors to D numbers. In this example since multiple decision makers are included so as to form a group decision making environment, we use the proportion of each evaluation to construct the D numbers. For example, for FM 1 on risk factor O, five decision makers respectively give evaluations M, M, M, MH, M, hence we can construct a D number $D(M) = 0.8$, $D(MH) = 0.2$. In terms of this means, the evaluations to failure modes are transformed to the form of D numbers, as shown in Table 8.

Table 8. The evaluations to failure modes in the form of D numbers.

Failure Mode	O	S	D
FM 1	$D(\{M\}) = 0.8,$ $D(\{MH\}) = 0.2$	$D(\{ML\}) = 0.6,$ $D(\{M\}) = 0.4$	$D(\{M\}) = 0.2,$ $D(\{ML\}) = 0.8$
FM 2	$D(\{H\}) = 0.4,$ $D(\{MH\}) = 0.6$	$D(\{H\}) = 0.8,$ $D(\{MH\}) = 0.2$	$D(\{M\}) = 0.8,$ $D(\{ML\}) = 0.2$
FM 3	$D(\{VH\}) = 0.8,$ $D(\{MH\}) = 0.2$	$D(\{MH\}) = 1$	$D(\{MH\}) = 0.6,$ $D(\{M\}) = 0.4$
FM 4	$D(\{M\}) = 0.8,$ $D(\{L\}) = 0.2$	$D(\{M\}) = 0.8,$ $D(\{ML\}) = 0.2$	$D(\{VL\}) = 0.6,$ $D(\{ML\}) = 0.4$
FM 5	$D(\{M\}) = 0.8,$ $D(\{ML\}) = 0.2$	$D(\{M\}) = 0.6,$ $D(\{MH\}) = 0.4$	$D(\{L\}) = 0.8,$ $D(\{ML\}) = 0.2$
FM 6	$D(\{M\}) = 0.4,$ $D(\{MH\}) = 0.4,$ $D(\{H\}) = 0.2,$	$D(\{H\}) = 1$	$D(\{L\}) = 0.6,$ $D(\{M\}) = 0.2,$ $D(\{VL\}) = 0.2$

Step 6: Rank the failure modes using the proposed multi-sensor information fusion method. At first, the fuzzy linguistic variables in Table 4 form a FOD $\Theta = \{VL, L, ML, M, MH, H, VH\}$. Each exclusive degree between elements in Θ has been obtained in Example 1. According to Equation (15), the non-exclusive degree of any pair of elements in 2^Θ can be easily obtained. Secondly,

for every failure mode the evaluations on O, S, and D can be fused based on the proposed combination operation in Definition 6. For FM 1, the aggregated evaluation is

$$\begin{aligned}D_1(\{M\}) &= 0.558, \\D_1(\{ML\}) &= 0.263, \\D_1(\{ML, M\}) &= 0.175, \\D_1(\{ML, M, MH\}) &= 0.004.\end{aligned}$$

For FM 2,

$$\begin{aligned}D_2(\{M, MH\}) &= 0.702, \\D_2(\{M, MH, H\}) &= 0.298.\end{aligned}$$

For FM 3,

$$\begin{aligned}D_3(\{MH\}) &= 0.943, \\D_3(\{M, MH\}) &= 0.057.\end{aligned}$$

For FM 4,

$$\begin{aligned}D_4(\{ML\}) &= 0.236, \\D_4(\{ML, M\}) &= 0.756, \\D_4(\{VL, L, ML\}) &= 0.008.\end{aligned}$$

For FM 5,

$$\begin{aligned}D_5(\{ML\}) &= 0.178, \\D_5(\{ML, M\}) &= 0.714, \\D_5(\{ML, M, MH\}) &= 0.043, \\D_5(\{L, ML, M\}) &= 0.065.\end{aligned}$$

For FM 6,

$$D_6(\{M, MH, H\}) = 1.000.$$

Thirdly, by using the PPT, we have: $BetP_1(\{ML\}) = 0.3516$, $BetP_1(\{M\}) = 0.6471$, $BetP_1(\{MH\}) = 0.0013$ for FM 1; $BetP_2(\{M\}) = 0.4504$, $BetP_2(\{MH\}) = 0.4504$, $BetP_2(\{H\}) = 0.0993$ for FM 2; $BetP_3(\{M\}) = 0.0286$, $BetP_3(\{MH\}) = 0.9714$ for FM 3; $BetP_4(\{VL\}) = 0.0026$, $BetP_4(\{L\}) = 0.0026$, $BetP_4(\{ML\}) = 0.6168$, $BetP_4(\{M\}) = 0.3780$ for FM 4; $BetP_5(\{L\}) = 0.0216$, $BetP_5(\{ML\}) = 0.5712$, $BetP_5(\{M\}) = 0.3928$, $BetP_5(\{MH\}) = 0.0144$ for FM 5; and $BetP_6(\{M\}) = 0.3333$, $BetP_6(\{MH\}) = 0.3333$, $BetP_6(\{H\}) = 0.3333$ for FM 6. Fourthly, these pignistic probabilities are then transformed to fuzzy aggregated evaluations according to Equations (21) and (22) which are

$$\begin{aligned}\tilde{F}_1 &= (3.2982, 4.2982, 4.6511, 5.6511), \\ \tilde{F}_2 &= (4.7482, 5.7482, 6.1986, 7.1986), \\ \tilde{F}_3 &= (4.9714, 5.9714, 6.9429, 7.9429), \\ \tilde{F}_4 &= (2.7480, 3.7454, 4.3648, 5.3648), \\ \tilde{F}_5 &= (2.8072, 3.8072, 4.3928, 5.3928), \\ \tilde{F}_6 &= (5.3333, 6.3333, 6.6667, 7.6667).\end{aligned}$$

These fuzzy aggregated evaluations are graphically shown in Figure 4. At last, in terms of the centroid defuzzification approach we can have $C_{\tilde{F}_1} = 4.4746$, $C_{\tilde{F}_2} = 5.9734$, $C_{\tilde{F}_3} = 6.4571$, $C_{\tilde{F}_4} = 4.0559$, $C_{\tilde{F}_5} = 4.1000$, $C_{\tilde{F}_6} = 6.5000$. Therefore, the risk ranking of all failure modes from high to low is $FM\ 6 \succ FM\ 3 \succ FM\ 2 \succ FM\ 1 \succ FM\ 5 \succ FM\ 4$. From the result, it is found that the failure mode with the highest risk is FM 6 and that having the lowest risk is FM 4.

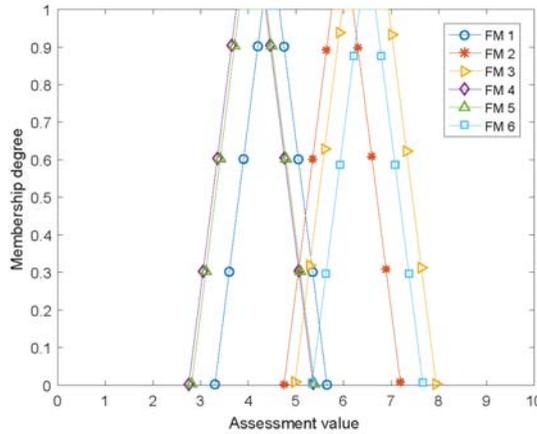


Figure 4. Graphically presentation of fuzzy aggregated evaluations.

The above steps have clearly shown the process of using the proposed model to do the risk evaluation in FMEA under fuzzy environment. Now we compare the result obtained by the proposed model with that from other method. In literature [28], Liu et al. dealt with the risk evaluation in FMEA with an extended VIKOR method. The results of the risk ranking are given in Table 9. In [28], the failure modes are ranked in terms of three indicators S, R, Q. By S, the failure modes with the highest and lowest risk are FM 3 and FM 4; By R, the failure modes having the highest and lowest risk are respectively FM 6 and FM 4; By Q, the failure modes with the highest and lowest risk are FM 3 and FM 4, respectively. Comparing the proposed model and the extended VIKOR method in [28], the ranking obtained by the proposed model is basically same with that of R. In addition, both the two methods have identified FM 4 is the failure mode of lowest risk. In addition, as a whole the failure modes can be classified into two groups by S, R, Q, and the first group which has higher risk is composed by FM 3, FM 6, FM 2, the second group having lower risk includes FM 1, FM 5, FM 4. By using the proposed model, we also obtain the same classification that FM 6, FM 3, FM 2 are in the group with higher risk and FM 1, FM 5, FM 4 constitute the group with lower risk. Through the above analysis and comparison, therefore it shows that the proposed model is effective for risk evaluation in FMEA.

Table 9. Risk ranking of failure modes by using the extended VIKOR method in [28].

	Failure Mode					
	FM 1	FM 2	FM 3	FM 4	FM 5	FM 6
By S	4	2	1	6	5	3
By R	5	3	2	6	4	1
By Q	5	3	1	6	4	2

6. Conclusions

In this paper, the risk evaluation of failure modes in FMEA has been studied in an environment involving fuzzy uncertainty. A novel model is proposed for the fuzzy risk evaluation in FMEA. Within the proposed model, a D numbers based multi-sensor information fusion method has been presented to evaluate potential failure modes and rank the risk levels of failure modes. Since the use of D numbers which is a new model of extending classical D-S theory, the non-exclusiveness between the evaluations of fuzzy linguistic variables has been taken into consideration in the proposed method.

Compared with some existing risk evaluation methods in FMEA, the proposed method overcomes the shortcomings of traditional RPN approach to some degrees and obtains comparable performances relative to other MCDM technologies used in FMEA. This study provides a new solution for the risk evaluation in FMEA under fuzzy environment and it is especially suitable for the case that contains non-exclusive fuzzy evaluations. In the future research, on one hand we hope to seek more practical applications with read data to analyze the proposed D numbers based multi-sensor information fusion method, on the other hand we will explore other technologies such as Physarum-inspired model [64] to improve the risk evaluation in FMEA.

Acknowledgments: The authors are grateful to anonymous reviewers for their useful comments and suggestions. The work is partially supported by National Natural Science Foundation of China (Grant No. 61703338, 61671384), Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016JM6018), Aviation Science Foundation (Program No. 20165553036), Fundamental Research Funds for the Central Universities (Program No. 31020170QD020).

Author Contributions: X.D. and W.J. conceived, designed, performed the experiments and analyzed the data; X.D. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stamatis, D.H. *Failure Mode and Effect Analysis: FMEA from Theory to Execution*; ASQ Quality Press: Milwaukee, WI, USA, 2003.
2. Liu, H.C.; Liu, L.; Lin, Q.L. Fuzzy failure mode and effects analysis using fuzzy evidential reasoning and belief rule-based methodology. *IEEE Trans. Reliab.* **2013**, *62*, 23–36.
3. Arabian-Hoseynabadi, H.; Oraee, H.; Tavner, P. Failure modes and effects analysis (FMEA) for wind turbines. *Int. J. Electr. Power Energy Syst.* **2010**, *32*, 817–824.
4. Wang, Y.M.; Chin, K.S.; Poon, G.K.K.; Yang, J.B. Risk evaluation in failure mode and effects analysis using fuzzy weighted geometric mean. *Expert Syst. Appl.* **2009**, *36*, 1195–1207.
5. Franceschini, F.; Galetto, M. A new approach for evaluation of risk priorities of failure modes in FMEA. *Int. J. Prod. Res.* **2001**, *39*, 2991–3002.
6. Zammori, F.; Gabbriellini, R. ANP/RPN: A multi criteria evaluation of the risk priority number. *Qual. Reliab. Eng. Int.* **2012**, *28*, 85–104.
7. Xiao, N.; Huang, H.Z.; Li, Y.; He, L.; Jin, T. Multiple failure modes analysis and weighted risk priority number evaluation in FMEA. *Eng. Fail. Anal.* **2011**, *18*, 1162–1170.
8. Braglia, M.; Frosolini, M.; Montanari, R. Fuzzy criticality assessment model for failure modes and effects analysis. *Int. J. Qual. Reliab. Manag.* **2003**, *20*, 503–524.
9. Liu, H.C.; You, J.X.; Li, P.; Su, Q. Failure mode and effect analysis under uncertainty: An integrated multiple criteria decision making approach. *IEEE Trans. Reliab.* **2016**, *65*, 1380–1392.
10. Liu, H.C. *FMEA Using Uncertainty Theories and MCDM Methods*; Springer: Singapore, 2016.
11. Chin, K.S.; Chan, A.; Yang, J.B. Development of a fuzzy FMEA based product design system. *Int. J. Adv. Manuf. Technol.* **2008**, *36*, 633–649.
12. Yang, Z.; Bonsall, S.; Wang, J. Fuzzy rule-based Bayesian reasoning approach for prioritization of failures in FMEA. *IEEE Trans. Reliab.* **2008**, *57*, 517–528.
13. Yeh, T.M.; Chen, L.Y. Fuzzy-based risk priority number in FMEA for semiconductor wafer processes. *Int. J. Prod. Res.* **2014**, *52*, 539–549.
14. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353.
15. Chin, K.S.; Wang, Y.M.; Poon, G.K.K.; Yang, J.B. Failure mode and effects analysis by data envelopment analysis. *Decis. Support Syst.* **2009**, *48*, 246–256.
16. Jee, T.L.; Tay, K.M.; Lim, C.P. A new two-stage fuzzy inference system-based approach to prioritize failures in failure mode and effect analysis. *IEEE Trans. Reliab.* **2015**, *64*, 869–877.
17. Chang, W.L.; Tay, K.M.; Lim, C.P. A new application of an evolving tree to failure mode and effect analysis methodology. In Proceedings of the International Conference on Neural Information Processing, Kuching, Malaysia, 3–6 November 2014; Springer: Cham, Switzerland, 2014; pp. 415–422.

18. Liu, H.C.; Liu, L.; Liu, N. Risk evaluation approaches in failure mode and effects analysis: A literature review. *Expert Syst. Appl.* **2013**, *40*, 828–838.
19. Waltz, E.; Llinas, J. *Multisensor Data Fusion*; Artech House: Boston, MA, USA, 1990.
20. Mitchell, H.B. *Multi-Sensor Data Fusion: An Introduction*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
21. Ji, L.; Yang, F.; Wang, X.; Chen, L. An uncertain information fusion method based on possibility theory in multisource detection systems. *Opt. Int. J. Light Electron Opt.* **2014**, *125*, 4583–4587.
22. Jiang, W.; Xie, C.; Zhuang, M.; Shou, Y.; Tang, Y. Sensor data fusion with Z-numbers and its application in fault diagnosis. *Sensors* **2016**, *16*, 1509, doi:10.3390/s16091509.
23. Sun, S.; Lin, H.; Ma, J.; Li, X. Multi-sensor distributed fusion estimation with applications in networked systems: A review paper. *Inf. Fusion* **2017**, *38*, 122–134.
24. Tian, T.; Sun, S.; Li, N. Multi-sensor information fusion estimators for stochastic uncertain systems with correlated noises. *Inf. Fusion* **2016**, *27*, 126–137.
25. Palmieri, F.; Ciunzo, D. Objective priors from maximum entropy in data classification. *Inf. Fusion* **2013**, *14*, 186–198.
26. Palmieri, F.; Ciunzo, D. Data fusion with entropic priors. In Proceedings of the 20th Italian Workshop on Neural Nets, WIRN, Frontiers in Artificial Intelligence and Applications, Vietri sul Mare, Italy, 27–29 May 2010; Volume 226, pp. 107–114.
27. Song, W.; Ming, X.; Wu, Z.; Zhu, B. A rough TOPSIS approach for failure mode and effects analysis in uncertain environments. *Qual. Reliab. Eng. Int.* **2014**, *30*, 473–486.
28. Liu, H.C.; Liu, L.; Liu, N.; Mao, L.X. Risk evaluation in failure mode and effects analysis with extended VIKOR method under fuzzy environment. *Expert Syst. Appl.* **2012**, *39*, 12926–12934.
29. Jiang, W.; Xie, C.; Zhuang, M.; Tang, Y. Failure mode and effects analysis based on a novel fuzzy evidential method. *Appl. Soft Comput.* **2017**, *57*, 672–683.
30. Deng, Y. D numbers: Theory and applications. *J. Inf. Comput. Sci.* **2012**, *9*, 2421–2428.
31. Deng, X.; Hu, Y.; Deng, Y.; Mahadevan, S. Supplier selection using AHP methodology extended by D numbers. *Expert Syst. Appl.* **2014**, *41*, 156–167.
32. Deng, X.; Hu, Y.; Deng, Y.; Mahadevan, S. Environmental impact assessment based on D numbers. *Expert Syst. Appl.* **2014**, *41*, 635–643.
33. Zhou, X.; Deng, X.; Deng, Y.; Mahadevan, S. Dependence assessment in human reliability analysis based on D numbers and AHP. *Nucl. Eng. Des.* **2017**, *313*, 243–252.
34. Dempster, A.P. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339.
35. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
36. Yang, F.; Wei, H. Fusion of infrared polarization and intensity images using support value transform and fuzzy combination rules. *Infrared Phys. Technol.* **2013**, *60*, 235–243.
37. Wang, X.; Yang, F.; Wei, H.; Ji, L. A risk assessment model of uncertainty system based on set-valued mapping. *J. Intell. Fuzzy Syst.* **2016**, *31*, 3155–3162.
38. Ford Motor Company. *Potential Failure Mode and Effect Analysis (FMEA) Reference Manual*; Ford Motor Company: Dearborn, MI, USA, 1988.
39. Sankar, N.R.; Prabhu, B.S. Modified approach for prioritization of failures in a system failure mode and effects analysis. *Int. J. Qual. Reliab. Manag.* **2001**, *18*, 324–336.
40. Deng, X.; Jiang, W. An evidential axiomatic design approach for decision making using the evaluation of belief structure satisfaction to uncertain target values. *Int. J. Intell. Syst.* **2017**, in press, doi:10.1002/int.21929.
41. Yang, Y.; Han, D. A new distance-based total uncertainty measure in the theory of belief functions. *Knowl.-Based Syst.* **2016**, *94*, 114–123.
42. Yager, R.R.; Alajlan, N. Maxitive Belief Structures and Imprecise Possibility Distributions. *IEEE Trans. Fuzzy Syst.* **2017**, *25*, 768–774.
43. Elmore, P.A.; Petry, F.E.; Yager, R.R. Geospatial Modeling Using Dempster-Shafer Theory. *IEEE Trans. Cybern.* **2017**, *47*, 1551–1561.
44. Deng, X.; Xiao, F.; Deng, Y. An improved distance-based total uncertainty measure in belief function theory. *Appl. Intell.* **2017**, *46*, 898–915.

45. Jiang, W.; Wang, S. An uncertainty measure for interval-valued evidences. *Int. J. Comput. Commun. Control* **2017**, *12*, 631–644.
46. Wu, H.; Siegel, M.; Stiefelbogen, R.; Yang, J. Sensor fusion using Dempster-Shafer theory [for context-aware HCI]. In Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IMTC/2002), Anchorage, AK USA, 21–23 May 2002; IEEE: Piscataway, NJ, USA, 2002; Volume 1, pp. 7–12.
47. Hégarat-Masclé, S.L.; Richard, D.; Otlé, C. Multi-scale data fusion using Dempster-Shafer evidence theory. *Integr. Comput.-Aided Eng.* **2003**, *10*, 9–22.
48. Deng, X.; Jiang, W.; Zhang, J. Zero-sum matrix game with payoffs of Dempster-Shafer belief structures and its applications on sensors. *Sensors* **2017**, *17*, 922, doi:10.3390/s17040922.
49. Xu, X.; Li, Z.; Li, G.; Zhou, Z. State Estimation Using Dependent Evidence Fusion: Application to Acoustic Resonance-Based Liquid Level Measurement. *Sensors* **2017**, *17*, 924, doi:10.3390/s17040924.
50. Su, X.; Mahadevan, S.; Han, W.; Deng, Y. Combining dependent bodies of evidence. *Appl. Intell.* **2016**, *44*, 634–644.
51. Deng, X.; Han, D.; Dezert, J.; Deng, Y.; Shyr, Y. Evidence combination from an evolutionary game theory perspective. *IEEE Trans. Cybern.* **2016**, *46*, 2070–2082.
52. Song, Y.; Wang, X.; Lei, L.; Yue, S. Uncertainty measure for interval-valued belief structures. *Measurement* **2016**, *80*, 241–250.
53. Deng, Y. Generalized evidence theory. *Appl. Intell.* **2015**, *43*, 530–543.
54. Jiang, W.; Zhan, J. A modified combination rule in generalized evidence theory. *Appl. Intell.* **2017**, *46*, 630–640.
55. Jiang, W.; Wang, S.; Liu, X.; Zheng, H.; Wei, B. Evidence conflict measure based on OWA operator in open world. *PLoS ONE* **2017**, *12*, e0177828.
56. Han, D.; Dezert, J.; Duan, Z. Evaluation of probability transformations of belief functions for decision making. *IEEE Trans. Syst. Man. Cybern. Syst.* **2016**, *46*, 93–108.
57. Smets, P.; Kennes, R. The transferable belief model. *Artif. Intell.* **1994**, *66*, 191–234.
58. Wang, N.; Liu, X.; Wei, D. A modified D numbers' integration for multiple attributes decision making. *Int. J. Fuzzy Syst.* **2017**, 1–12, doi:10.1007/s40815-017-0323-0.
59. Deng, X.; Jiang, W. Exploring the combination rules of D numbers from a perspective of conflict redistribution. In Proceedings of the 20th International Conference on Information Fusion (Fusion), Xi'an, China, 10–13 July 2017; pp. 1–6, doi:10.23919/ICIF.2017.8009696.
60. Yager, R.R.; Filev, D.P. Induced ordered weighted averaging operators. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **1999**, *29*, 141–150.
61. Yager, R.R. The power average operator. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2001**, *31*, 724–731.
62. Jiang, W.; Wei, B.; Zhan, J.; Xie, C.; Zhou, D. A visibility graph power averaging aggregation operator: A methodology based on network analysis. *Comput. Ind. Eng.* **2016**, *101*, 260–268.
63. Ebrahimnejad, S.; Mousavi, S.; Tavakkoli-Moghaddam, R.; Hashemi, H.; Vahdani, B. A novel two-phase group decision making approach for construction project selection in a fuzzy environment. *Appl. Math. Model.* **2012**, *36*, 4197–4217.
64. Xu, S.; Jiang, W.; Deng, X.; Shou, Y. A modified Physarum-inspired model for the user equilibrium traffic assignment problem. *Appl. Math. Model.* **2017**, in press, doi:10.1016/j.apm.2017.07.032.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Modeling of BN Lifetime Prediction of a System Based on Integrated Multi-Level Information

Jingbin Wang ¹, Xiaohong Wang ¹ and Lizhi Wang ^{2,3,*}

¹ School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China; wangjingbin@buaa.edu.cn (J.W.); wxhong@buaa.edu.cn (X.W.)

² Unmanned System Institute, Beihang University, Beijing 100191, China

³ Key Laboratory of Advanced Technology of Intelligent Unmanned Flight System of Ministry of Industry and Information Technology, Beihang University, Beijing 100191, China

* Correspondence: wanglizhi@buaa.edu.cn; Tel.: +86-138-1043-8269

Received: 11 August 2017; Accepted: 13 September 2017; Published: 15 September 2017

Abstract: Predicting system lifetime is important to ensure safe and reliable operation of products, which requires integrated modeling based on multi-level, multi-sensor information. However, lifetime characteristics of equipment in a system are different and failure mechanisms are inter-coupled, which leads to complex logical correlations and the lack of a uniform lifetime measure. Based on a Bayesian network (BN), a lifetime prediction method for systems that combine multi-level sensor information is proposed. The method considers the correlation between accidental failures and degradation failure mechanisms, and achieves system modeling and lifetime prediction under complex logic correlations. This method is applied in the lifetime prediction of a multi-level solar-powered unmanned system, and the predicted results can provide guidance for the improvement of system reliability and for the maintenance and protection of the system.

Keywords: multi-level system; lifetime prediction; Bayesian Networks; multi-sensor information integration; complex logical correlation

1. Introduction

Life expectancy is of great significance to ensure safe and reliable operation of products, and demands for superior product performance are increasing. Devices and components with multiple lifetime characteristics are associated by complex logic to form a multi-level, complex system whose state features are difficult to quantify and predict directly. Therefore, the integration of multi-sensor information is a necessary path to achieving lifetime prediction for such systems. In fact, research on the integration of multi-sensor information is one of the trends in comprehensive utilization of data and monitoring technology innovation in the international research community. Through signal processing from multiple sensors, combined with optimization theory [1], pattern recognition [2], filtering technique [3,4], neural networks [5], artificial intelligence [6], etc., the goal is to achieve the correlation and combination of information, and provide more accurate monitoring and evaluation of products.

In order to effectively integrate equipment information provided by sensors in a complex, multi-level system to achieve evaluation and prediction of the lifetime of a system, the following requirements must be met: first, sensor data are required to carry out evaluation at the equipment level for designing prediction models. Second, the prediction framework that can integrate these low-level prediction models must be provided. A variety of prediction methods have been developed at the equipment level, including statistical methods that are based on Wiener diffusion [7], the gamma process [8], and the Markov process [9], as well as artificial intelligence methods such as data-driven neural networks [10,11] and filtering techniques [12,13], and other methods based on physical

failure [14]. For instance, Nguyen considered the predictive reliability, dependencies of components and the hierarchy configuration to study the multi-level system [15]. Zheng combined unscented Kalman Filter and Relevance Vector Regression to predict the remaining useful life of lithium-ion batteries [16]. Hamed compared the stochastic simulation and the inverse First Order Reliability Method in remaining useful life prediction under different forms of uncertainty [17]. Actually, owing to evolving sensor technology, studies on prediction models for lower level objects and simple products are prevalent, and data-based lifetime prediction technology is relatively mature. However, the construction of the system prediction framework should take into account the uniform quantification of lifetime indicators and the expression of complex logic. Coolen extended the sensor status information obtained by a sensor to the probability level, and carried out the uncertainty prediction research for many types of components [18]. Jackson and Mosleh also studied the predictive modeling of uncertainty for multilayer overlapping data in complex systems [19]. The above studies affirm the advantage of a probabilistic description of complex system states and logical association uncertainties. On the other hand, George [20] and Meineri [21] adopted directed graphs in their research on complex system topology modeling and multi-factor complex correlation, which inspired the idea of modeling logical relationships of multi-level system interactions.

The Bayesian network (BN), proposed by Pearl [22], is a reasoning model based on Bayesian theory and graph theory that integrates probability and graphitization. It can be applied to make up the deficiency of traditional reliability analysis method. For an example, with the Fault Tree Analysis (FTA) method it is difficult to model systems with polymorphism problems, random uncertainties and dependent events, and some calculations are impossible to achieve due to the computation complexity under the disjointed algorithm scenario, while BN is able to describe the multi-state, uncertainties and correlations to conduct two-way inference due to probability basis, conditional independent principle and information integration skills, as well as generate results conveniently according to the available FT and simplify modeling problems. Therefore BN has a great advantage in complex system modeling applications such as disease diagnosis, financial risk analysis, and wireless sensor network and system reliability analysis, and provides the application basis for integration of multi-level information [23], which can be used to carry out the construction of lifetime prediction models based on state probability. A BN qualitative network topology and quantitative conditional probability description are endowed with strong ability to express large complex systems that have a large number of subsystems [24], which is applicable for the modeling of complex relationships such as system reliability diagrams and fault trees that are difficult to describe using traditional models [25,26]. At present, BN is widely used in the field of reliability, for system fault diagnosis [27,28], safety analysis [29], and optimization of maintenance strategy [30]. For instance, Cai et al. proposed a BN-based data-driven fault detection and diagnosis methodology which is equipped with a good toleration for sensor noise and bias of PMSM drive system [31]. Petek conducted an evaluation of failure conditional probability with a multi-level system by BN method [32]. Hu et al. developed an integrated safety prognosis model with BN to study the propagation mechanisms of faults in complex system [33]. However, there are few reports on the prediction method for a complex, multi-level system [34]. Kabir et al. investigated the failure prediction method of water mains with BN by considering the uncertainties from multi-source data and human interpretations with different credibility [35]. Therefore, system lifetime prediction based on BN to achieve the integration of equipment information from multiple sensors has substantial research value and exploration significance.

In order to solve the problem of lifetime prediction of multi-level systems under complex logic relationships and guide the design, improvement, and maintenance of systems, this paper studies the BN lifetime prediction method based on the integration of multi-level information. The system life expectancy is estimated by considering different failure mechanisms and the logical association of its mutual coupling, and the integration of equipment information with different lifetime characteristics. This paper is organized as follows: in Section 2, we present an overview of the basic concept, construction, and estimation algorithm of the BN model. In Section 3, we describe the modeling

process of BN lifetime prediction and its solution method based on the prediction model at equipment level. In Section 4, we present the verification of the prediction model and in Section 5 a case study of a solar unmanned system. Finally, a summary and outlook are given in Section 6.

2. Overview of BN Model

A BN is a graphical network model based on the probabilistic reasoning of Bayesian theory. It consists of a Directed Acyclic Graph (DAG) and a Conditional Probability Table (CPT). The former is a graphical structure composed of directed edges that connect the node variable $X = \{X_1, X_2, \dots, X_n\}$ based on causality. The latter is the quantitative expression of the logical relationship of variables. The directed edge in a DAG always has the parent node pointing to the child node, while the variable with no parent node is the root node and the variable with no child node is the leaf node, and the rest are the intermediate nodes.

The probability calculation of a BN is based on the conditional independence assumption, i.e., that the probability of the child node depends only on the parent node and is independent from the other child nodes of the parent node, as shown in Equation (1). Therefore, when solving the probability of a multi-node joint, a BN only needs to consider the correlation of variables, thus reduces the solving complexity:

$$P(X_i|X_{pi}, X_{pai}) = P(X_i|X_{pi}) \tag{1}$$

where X_{pi} is the parent node of node X_i and X_{pai} the child node of X_{pi} other than X_i . Applying conditional independence to chain rules enables computation of the joint probability, as follows:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i|X_{pi}) \tag{2}$$

The following three elements must be completed for a BN construction:

1. Determine node variables and their interpretation.
2. Create a DAG with a directed edge connecting node variables.
3. Create a CPT for non-root nodes.

The CPT of node G is established for the seven-node DAG of Figure 1 according to the series logic, as shown in Table 1.

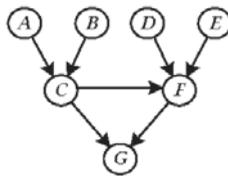


Figure 1. Seven-node DAG of system G.

Table 1. CPT of node G.

C		F		G	
F	T	F	T	F	T
1	0	1	0	1	0
1	0	0	1	0	1
0	1	1	0	0	1
0	1	0	1	0	1

In the precision reasoning and approximate reasoning algorithm, the joint tree (JT) algorithm based on clique tree propagation is widely used because of its unique advantages of high search efficiency, capability of returning logarithmic results, and dual-channel transmission. The solution is shown in Figure 2. The JT is obtained by the steps of the transformation and triangulation of the DAG. After initialization, the nodes can absorb the information and update the distribution function ϕ_C of the partition nodes to achieve information transmission. When the single-transfer process of the JT information satisfies the globally consistent steady state, the distribution of V can be obtained according to $P(V) = \sum_{C\{V\}} \phi_C$ for any desired variable V . When new evidence e is added, the conditional probability distribution of the variable V is solved as follows:

$$P(V|e) = \frac{P(V, e)}{P(e)} = \frac{P(V, e)}{\sum_V P(V, e)} \tag{3}$$

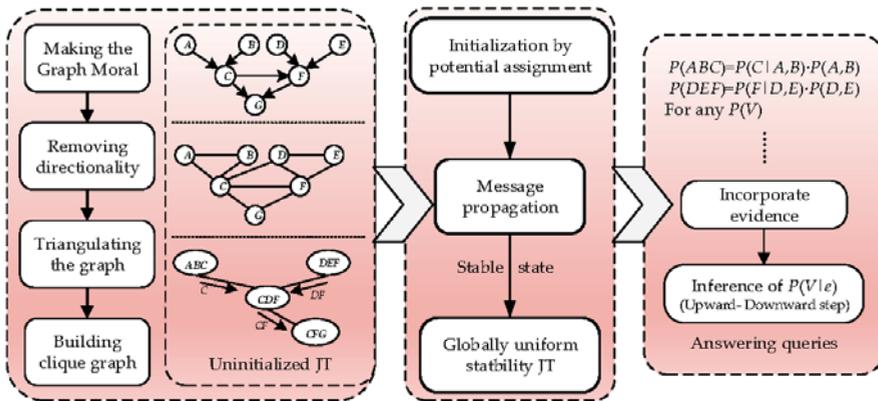


Figure 2. Algorithm and build thought of JT.

3. Lifetime Prediction Method of Multi-level System Based on BN

3.1. Prediction Model at Equipment Level

From the prediction tools and means, the prediction model at the equipment and component levels can be divided into three categories: stochastic process based, data based, and physical failure based. The optimal prediction method is chosen to establish the lifetime prediction model for root nodes of the BN, and the lifetime prediction of intermediate nodes and leaf nodes can then be deduced. Therefore, the soundness and accuracy of the model and parameters of the equipment can affect the prediction of higher-level nodes.

The system of a complex product often involves performance degradation and accidental failure. Here, the equipment whose performance degradation obeys the Wiener process is taken as an example to devise a prediction model. Assuming that the performance parameter W is a key indicator of product lifetime and is sensitive to stress S , the parameter then follows the Wiener degradation process as follows:

$$W(t) = \mu(s) \cdot t + \sigma \cdot B(t) + W_0 \tag{4}$$

where $W(t)$ is the product performance at time t , and $\mu(s)$ is the drifting coefficient reflecting the performance degradation rate, which is a function of stress and time. In an accelerated model, $\mu(s) = \exp[\beta_0 + \beta\varphi(s)]$. Constant σ is the diffusion coefficient that is irrelevant with respect to environment and time. $B(t) \sim N(0, t)$ is the standard Brownian motion and W_0 is the initial value of the parameter.

The degradation amount within the time Δt from the properties of the Wiener process is $\Delta W \sim N(\mu(s)\Delta t, \sigma^2\Delta t)$. L is defined as the failure threshold of performance W , and then the time t' that the performance parameter value first passed through L satisfies the inverse Gaussian distribution. The distribution function is the unreliability function of the product, and the corresponding probability density function is given by:

$$f(t; W_0, L) = \frac{L - W_0}{\sqrt{2\pi\sigma^2 t^3}} \exp\left\{-\frac{[(L - W_0) - \mu(s)t]^2}{2\sigma^2 t}\right\} \quad (5)$$

The corresponding reliability function is the prediction model of equipment lifetime, as follows:

$$R(t) = \Phi\left(\frac{L - W_0 - \mu(s)t}{\sigma\sqrt{t}}\right) - \exp\left\{-\frac{2\mu(s)L}{2\sigma^2}\right\} \Phi\left(\frac{L - W_0 + \mu(s)t}{\sigma\sqrt{t}}\right), \quad (6)$$

where $\Phi(*)$ is the cumulative distribution function of the standard normal distribution.

Other prediction methods and models based on the data and physical failure can also provide the corresponding prediction information of product lifetime under the timescale satisfying prediction accuracy.

3.2. BN Prediction Modeling and Inference at System Level

Differences exist in the lifetime characteristics of different products. Therefore, the state probability of the unified lifetime based on quantitative indicators is used in this paper. For example, for the case of accidental hardware failure, reliability and cumulative probability of failure can describe the probability in the “normal” and “fault” states. However, for a degradation mechanism in which the lifetime characteristics cannot be directly described by the reliability, the complete state set must be customized before modeling to describe the “intact degree” and “failure degree” as two opposing events, to ensure that the sum of the state probability is 1. For ease of expression, in this work we use \mathcal{R} to characterize the “intact” state of all nodes, with corresponding probability R , and represent the corresponding “failure” state with \mathcal{F} , with a probability of F .

The lifetime prediction sequence of state probability reflecting lifetime information given by each equipment prediction model is taken as the prior probability of the corresponding root node in the system BN, and the probability of unknown nodes is deduced to achieve the integration and prediction of the lifetime information from the multi-sensor (process shown in Figure 3). The specific model construction proceeds via the following steps:

1. Obtain expert knowledge and structure and function information of similar products. Analyze the failure mode and mechanism through the system function-level method. Fault tree analysis is used for key faults to determine the equipment and mechanisms that affect system lifetime.
2. Deploy sensors for key performance parameters of each device. Collect and process data.
3. Based on the analysis of sensor data, the prediction model is established for each piece of equipment, and the prediction value of lifetime-related state probability is given.
4. Combined with the system failure mechanism, the prediction value of state probability for the equipment involved is used as the prior probability for the root node.
5. Combine system logic to form a DAG, and establish a non-root node CPT.
6. The JT estimation algorithm is used to solve the joint probability of relevant nodes, to update the conditional probability values of each node, and to achieve the deduction of state probability of system nodes to complete the system prediction.
7. The BN prediction model is still applicable, along with the updating of sensor data and correction of prediction model at equipment level. If the failure mechanism changes, the DAG and CPT should be corrected for the updated logical relationship; proceed to Step 3.

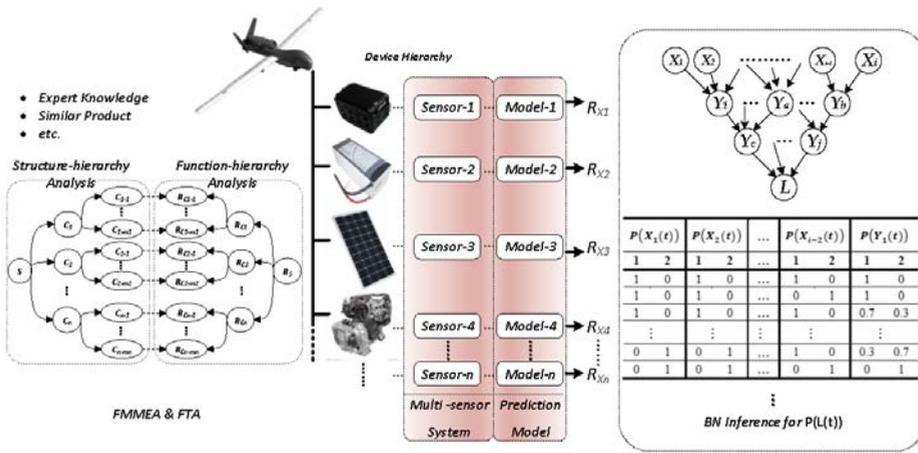


Figure 3. Modeling process of BN prediction at system level based on integrated multi-sensor information.

In the above BN model that interprets node variable with state probability, $S_{X_i}^{a_i}(t)$, $S_{Y_j}^{b_j}(t)$, and $S_L^k(t)$ are used to represent the a_i , b_j , k th state of root node X_i ($i = 1, 2, \dots, p$), the intermediate node Y_j ($j = 1, 2, \dots, q$), and the leaf node L at time t respectively, i.e., $a_i = b_j = k = 1, 2$.

For a number of p root nodes, the probability of the state \mathcal{R} is solved by the prediction model for each respective piece of equipment and discretized according to the unsupervised equal-width interval method with a time sequence to achieve the state prediction in the future T time; that is, the $p \times n$ -order state probability prediction matrix:

$$R_{p \times n} = \begin{pmatrix} R_{1,1} & \cdots & R_{1,n} \\ \vdots & \ddots & \vdots \\ R_{p,1} & \cdots & R_{p,n} \end{pmatrix},$$

with a certain time sequence $T = (t + \zeta, t + 2\zeta, \dots, t + n\zeta)$. The elements $R_{i,\tau} = R(X_i(\tau))$ ($i = 1, 2, \dots, p$; $\tau = 1, 2, \dots, n$) represent the probability of the device X_i to be in the state R . If ζ is taken as the unit time, the probability set $R(t) = (R_{1,t}, R_{2,t}, \dots, R_{p,t})^T$ corresponds to the probability of p nodes to be in set X at state \mathcal{R} . Correspondingly, the probability of occurrence of state F is given by $F(t) = 1 - R(t)$.

When the abovementioned multi-sensor information is used to deduce the lifetime prediction based on the BN, the probability of the root node is firstly assigned according to the probability prediction matrix. The probability of the root node X_i at time t is $P(S_{X_i}(t)) = R(X_i(t))$, $i = 1, 2, \dots, p$. For the solution of the state probability of intermediate nodes, it is assumed that the parent-node set $X = \{X_1, X_2, \dots, X_i\}$ exists for the node Y_j . According to the assumption of independent conditions, the probability prediction of the intermediate nodes at time t can be solved based on:

$$\begin{aligned} P(Y_j(t)) &= \sum_X P(S_{Y_j}(t), S_X(t)) \\ &= \sum_X P(S_{Y_j}^2(t) | S_X(t)) P(S_{X_1}(t)) \cdots P(S_{X_i}(t)), \end{aligned} \tag{7}$$

The complexity of logical associations of the BN will increase with the number of nodes contained. Based on the advantages of conditional independence, the complex topological equivalence can be divided into simple structures, and the construction and inference of three kinds of basic structural prediction models are taken as examples as follows:

- (1) The parent node set $X = \{X_1, X_2, X_3\}$ of node Y_j contains only the root node (Figure 4).

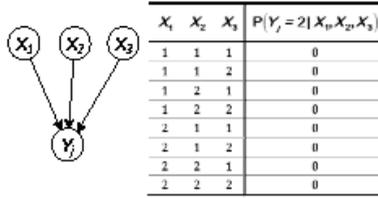


Figure 4. Logical structure in which the parent node is the root node.

Each state of the parent node is independent of each other at any time, and then the predicted probability of Y_j in state \mathcal{R} at time t is as follows:

$$\begin{aligned}
 R(Y_j(t)) &= \sum_{X_1, X_2, X_3} P(S_{Y_j}(t), S_{X_1}(t), S_{X_2}(t), S_{X_3}(t)) \\
 &= \sum_{X_1, X_2, X_3} P(S_{Y_j}^2(t) | S_X(t)) P(S_{X_1}(t)) P(S_{X_2}(t)) P(S_{X_3}(t)) \\
 &= P(S_{Y_j}^2(t) | S_{X_1}^2(t), S_{X_2}^2(t), S_{X_3}^2(t)) P(S_{X_1}^2(t)) P(S_{X_2}^2(t)) P(S_{X_3}^2(t)).
 \end{aligned}
 \tag{8}$$

- (2) The parent node set $X = \{X_a, Y_1\}$ of node Y_j contains both the root node and the intermediate node (Figure 5).

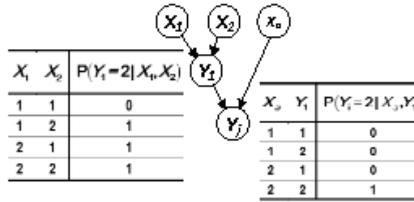


Figure 5. Logical structure in which the parent node contains both the intermediate and root nodes.

The predicted probability of node Y_j at time t can be solved according to:

$$\begin{aligned}
 R(Y_j(t)) &= \sum_{Y_1, X_a} P(S_{Y_j}^2(t) | S_{Y_1}^2(t), S_{X_a}^2(t)) P(S_{X_a}^2(t)) \sum_{X_1, X_2} P(S_{Y_1}^2(t) | S_{X_1}(t), S_{X_2}(t)) \cdot P(S_{X_1}(t)) P(S_{X_2}(t)) \\
 &= P(S_{Y_j}^2(t) | S_{Y_1}^2(t), S_{X_a}^2(t)) P(S_{X_a}^2(t)) \cdot [P(S_{Y_1}^2(t) | S_{X_1}^2(t), S_{X_2}^2(t)) P(S_{X_1}^2(t)) P(S_{X_2}^2(t)) \\
 &\quad + P(S_{Y_1}^2(t) | S_{X_1}^1(t), S_{X_2}^2(t)) P(S_{X_1}^1(t)) P(S_{X_2}^2(t)) + P(S_{Y_1}^2(t) | S_{X_1}^2(t), S_{X_2}^1(t)) P(S_{X_1}^2(t)) P(S_{X_2}^1(t))].
 \end{aligned}
 \tag{9}$$

- (3) There is a case in which the same parent node points to multiple child nodes (see Figure 6).

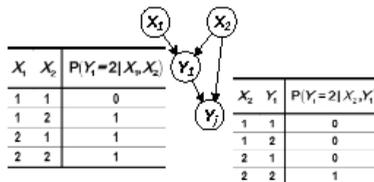


Figure 6. Logical structure in which the same parent node points to multiple subnodes.

The system model that has interconnected logical correlation mostly has network structure. The same parent node pointing to multiple sub-nodes is the basis for the composition of the network structure. This situation can be predicted according to:

$$\begin{aligned}
 R(Y_j(t)) &= \sum_{X_2, Y_1} P(S_{Y_j}^2(t)|S_{X_2}(t), S_{Y_1}(t)) \sum_{X_1, X_2} P(S_{Y_1}(t)|S_{X_1}(t), S_{X_2}(t)) \cdot P(S_{X_1}(t))P(S_{X_2}(t)) \\
 &= P(S_{Y_j}^2(t)|S_{X_2}^2(t), S_{Y_1}^2(t)) \cdot [P(S_{Y_1}^2(t)|S_{X_1}^2(t), S_{X_2}^2(t))P(S_{X_1}^2(t))P(S_{X_2}^2(t)) \\
 &\quad + P(S_{Y_1}^2(t)|S_{X_1}^1(t), S_{X_2}^2(t))P(S_{X_1}^1(t))P(S_{X_2}^2(t))] .
 \end{aligned}
 \tag{10}$$

Based on the probability of the root node and intermediate node, the predicted probability of the leaf node in state R can be further solved according to:

$$\begin{aligned}
 R_L(t) &= \sum P(S_{X_1}(t), S_{X_2}(t), \dots, S_{X_p}(t), S_{Y_1}(t), S_{Y_2}(t), \dots, S_{Y_q}(t), S_L^2(t)) \\
 &= \sum_{Pa(L)} P(S_L^2(t)|S_{Pa(L)}(t)) \cdot \sum_{Pa(Y_1)} P(S_{Y_1}(t)|S_{Pa(Y_1)}(t)) \cdots \\
 &\quad \sum_{Pa(Y_q)} P(S_{Y_q}(t)|S_{Pa(Y_q)}(t)) \cdots P(S_{X_1}(t)) \cdots P(S_{X_p}(t)) ,
 \end{aligned}
 \tag{11}$$

where $Pa(*)$ is the parent node of the node “(*)”.

According to the above formula, the JT estimation algorithm traverses the DAG, and the state and lifetime of the system node L can be predicted. By the probability prediction matrix $R_{p \times n} = (R(t), R(t + \zeta), R(t + 2\zeta), \dots, R(t + 2\zeta))$ of the root node, the corresponding prediction sequence of probability at system level will be obtained to achieve continuous prediction of the lifetime.

4. Modeling, Simulation, and Verification of System-Level BN Prediction

The seven-node DAG shown in Figure 1 contains the three basic structures described above, and the BN prediction method is simulated and verified as an example. The three-tier system G consists of parallel subsystems C and F , comprising four devices A, B, D , and E . In the subsystem $C(F)$, $A (D)$ and $B (E)$ conduct different functions, respectively, which are in series logic. In addition, according to the early collection of product information, the logic that “if the state of subsystem C is abnormal, there is a 60% probability of failure for subsystem F ” exists between the subsystems. The key performance parameters of each piece of equipment were determined, monitoring devices were arranged, and the sensor signals were extracted, processed, and analyzed. The lifetime prediction model of each piece of equipment was then obtained.

- (1) The lifetime characteristics of equipment A and B meet the accidental failure, while D and E meet the degradation failure, and the corresponding prediction models are shown in Table 2.

Table 2. Prediction model (1) of four pieces of equipment.

Node	Description of Prediction Model	Lifetime Prediction Model
A	Exponential distribution	$R_A(t) = \exp(-t/3000)$
B	Weibull distribution	$R_B(t) = \exp[-(t/2300)]^{1.5}$
D	Wiener degradation process; Drift parameters are selected as the Arrhenius model	$R_D(t) = -\exp\left[\frac{2d(s)(L-Y_0)}{\sigma^2}\right] \Phi\left(-\frac{L-Y_0+d(s)t}{\sigma\sqrt{t}}\right) + \Phi\left(\frac{L-Y_0-d(s)t}{\sigma\sqrt{t}}\right)$ $(Ea_D = 0.473, A_D = 4.7 \times 10^5; Y_{0-D} = 5, L_D = 10, \sigma_D = 0.01)$
E	Degradation under cyclic stress	$R_E(t) = \Phi\left(\left(-0.0001(t/10)^3 + 200 - 80\right)/\sqrt{5}\right)$

Based on the prediction model above, the BN model with both accidental failure and degeneration failure is established to predict the system lifetime of the next 2000 h. The state probability of system G can be solved by:

$$\begin{aligned}
 R_G(t) &= \sum P(S_A(t), S_B(t), S_C(t), S_D(t), S_E(t), S_F(t), S_G^2(t)) \\
 &= \sum_{C,F} P(S_G^2(t)|S_C(t), S_F(t)) \cdot \sum_{A,B} P(S_C(t)|S_A(t), S_B(t)) \\
 &\quad \sum_{C,D,E} P(S_F(t)|S_D(t), S_E(t), S_C(t))P(S_A(t))P(S_B(t))P(S_D(t))P(S_E(t)).
 \end{aligned}
 \tag{12}$$

Due to the efficient two-way accurate-inference ability, “engine = jtree _ inf _ engine (bnet)” of the Bayes Net Toolbox (BNT) for MATLAB developed by Murphy [36] is used to conduct inference, and the probability of each group of parent and child nodes in state R is shown in Figure 7.

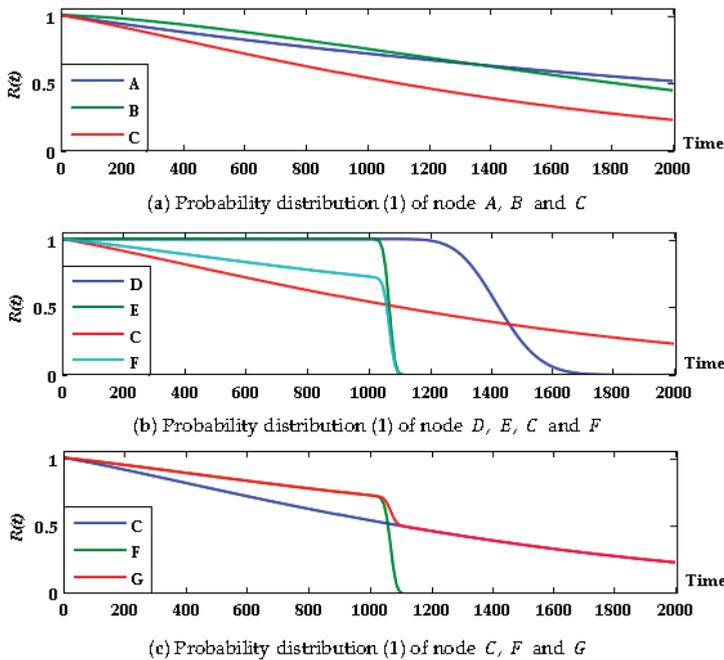


Figure 7. Probability distribution (1) of parent and child nodes at state R for G system group.

The results show that in the first 1000 h the subsystem F shows better performance than C, and the state probability of system G is the same as that of the former. After 1030 h, however, with the performance degradation, the failure probability of F suddenly increases, and the probability of system G being in state R decreases. After 1090 h, subsystem F completely failed, and the system state was completely determined by subsystem C. Therefore, the prediction curve trends of C and G after this point are basically the same. The lifetime prediction of the system is estimated and evaluated: the median life of the system $t_{0.5} = 1098$ h; if the system in good condition has a probability threshold of 0.45, the remaining life is approximately 1226 h.

In addition, the performance degradation simulation curve of E (as shown in Figure 8) shows that from 1010 to 1090 h the performance parameters of all 15 samples degenerated below the threshold of 80, and that the 15 samples of device D did not exceed the performance parameter threshold in the first 1200 h (a simulated degradation curve is shown in Figure 9). Through the analysis of

the subsystem *F*, we can see that the main reason for the sudden deterioration of the system is the performance degradation of device *E* under cyclic stress, which caused it to exceed the performance threshold with considerable probability.

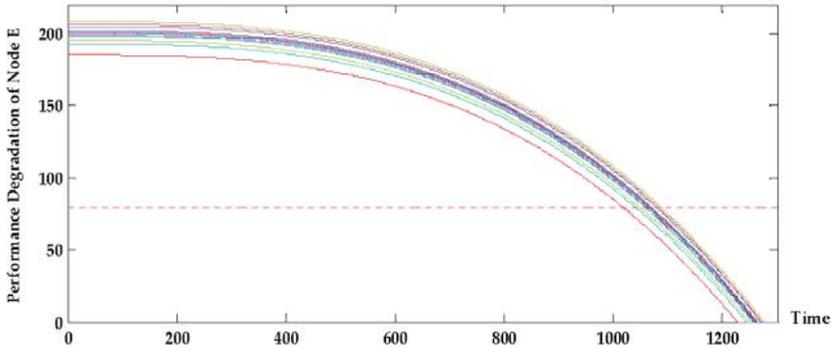


Figure 8. Simulation curve of performance degradation of device *E*.

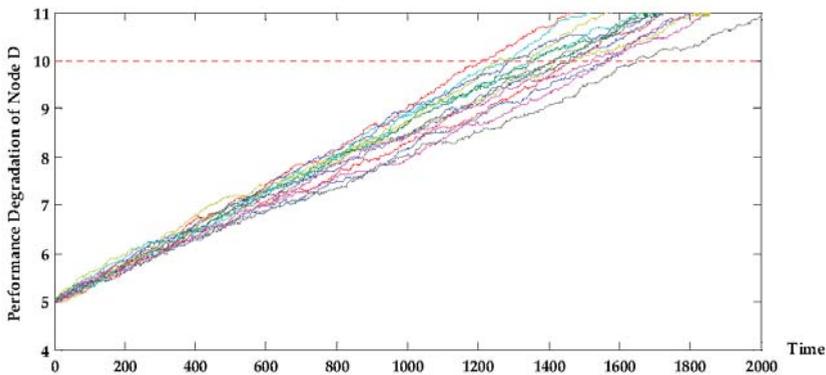


Figure 9. Simulation curve of performance degradation of device *D*.

The dotted lines parallel to the horizontal axis of Figures 8 and 9 represent the device performance threshold.

- (2) For all equipment that is in degradation failure mode, the corresponding prediction model and parameters are shown in Table 3.

Table 3. Prediction model (2) of four pieces of equipment ¹.

Node	Description of Prediction Model	Lifetime Prediction Model
<i>A'</i>	Gamma process; Scale parameters are selected as the Arrhenius model	$R_{A'}(t) = \Gamma(v(s)t, L/u) / \Gamma(v(s)t)$ ($Ea_{A'} = 0.453, A_{A'} = 3.9 \times 10^5; Y_{0-A'} = 0, L_{A'} = 12, u_{A'} = 2.3$)
<i>B'</i>	Wiener process	($Ea_{B'} = 0.473, A_{B'} = 6.7 \times 10^5; Y_{0-B'} = 6, L_{B'} = 12, \sigma_{B'} = 0.032$)

¹ Prediction models of *D* and *E* are the same as in Table 3.

The state probability prediction curve of each (sub-) system is obtained and shown in Figure 10.

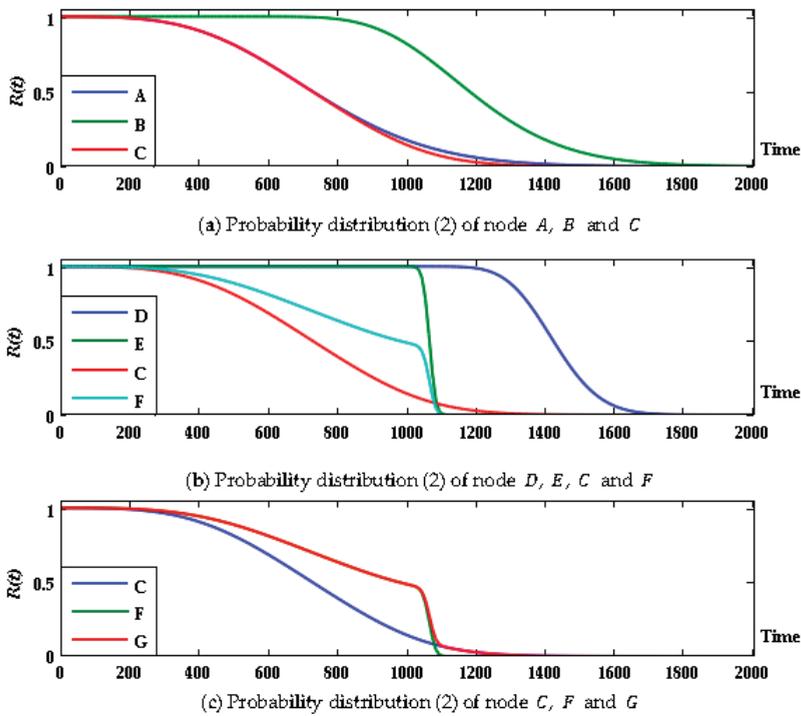


Figure 10. Probability distribution (2) of parent and child nodes at state R for G system group.

Compared to the predicted result above, a system whose devices are all in degradation failure mode has a low probability of staying intact at 1030 h, specifically, only 0.622. However, the state probability predicted above at 1030 h is 0.712, and the working state was maintained with a probability of 0.46 until 1200 h. With a probability threshold of 0.45, the remaining life of the system will be less than 1064 h. The main reason for the change of system state is subsystem C, in which the performance parameter of device A follows the Gamma degradation process (Figure 11). The probability of device A to be at state 2 is reduced to 0.5224 at 1000 h, making the system state significantly degraded; after this prediction point, the performance degradation of device E in subsystem F is deteriorated, and the system fails at a faster rate. On one hand, subsystem C has direct effect on the state of system G. On the other hand, the state of G is indirectly affected through the correlation with subsystem F. Therefore, the degradation rate of a system state after integration is faster.

It can be verified through the above simulation that a BN can achieve the integration of lifetime information from multiple sensors in a multi-level system when the equipment grade product experiences accidental failure or (and) degradation failure, which reasonably predicts system lifetime from the perspective of state probability, and shows good compatibility and integration for a variety of low-level prediction models.

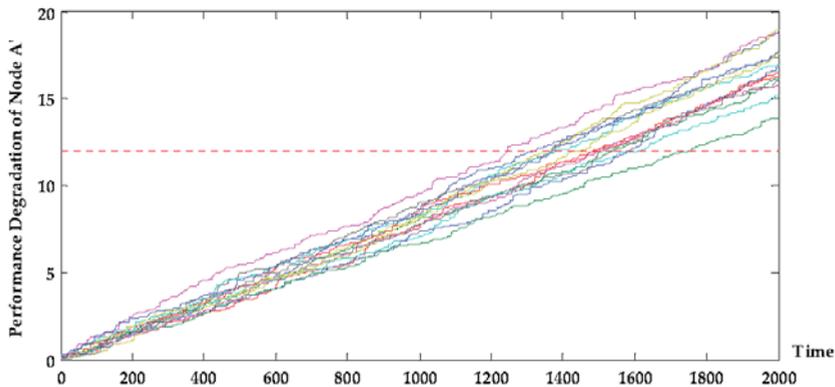


Figure 11. Simulation curve of performance degradation of equipment A'.

5. Application Case

5.1. BN Lifetime Prediction of an Unmanned Solar-Powered System

An unmanned system is a typical multi-level system with complex structure and function logic. With the air-ground collaborative work requirements, the system achieves its functional requirements and sharing of information resources layer by layer. At the same time, under environmental influences such as temperature, humidity, vibration, and interference the unit with the degradation property and that with the accidental failure together affect the system state and lifetime. In this paper, a solar-powered unmanned aerial vehicle (UAV) system was chosen as an example to carry out lifetime prediction based on a BN.

After the system is simplified, the hardware is mainly composed of an airborne system, a data link, and a ground system. The energy system inside the airborne system must provide power support for the normal operation of the navigation system, flight control system, power system, communication system, etc., and it is subject to the state of the battery management system and the solar panel. Any abnormality can cause the energy system to degrade or even fail to function properly.

The navigation system contains the master and lead. A flight control computer subsystem for integrated task management, a flight control computer subsystem, and a servo action system are necessary. In addition to the airborne terminal, the airborne communication system must go through the data link to achieve the transmission and exchange of information with ground communication equipment. In the ground subsystem, the power supply system is the energy supply system for flight operation, the console, communication equipment, and other equipment. For the console, the requirement is to simultaneously display and control functions. This BN model fully accounts for the relevance of intermediate nodes. In these subsystems, degradation failure exists in the solar cell and UAV body material, which is modeled with Wiener process and Gamma process, respectively, to establish the performance degradation for lifetime prediction. Specifically, the system DAG is shown in Figure 12, and the 61 nodes contain 37 device-level nodes.

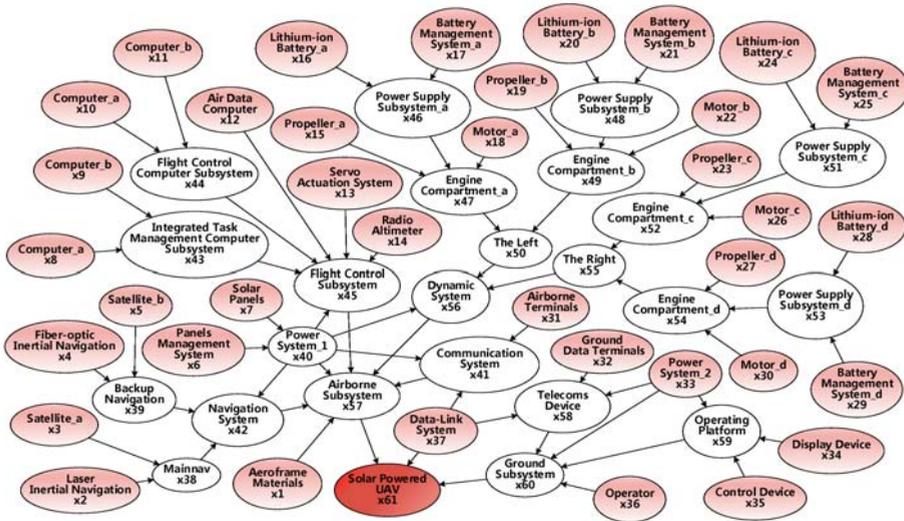


Figure 12. DAG of a solar-powered unmanned system.

At the same time, considering the characteristics of photovoltaic energy, a special flight profile is used: To ensure the full storage of the solar panels, before every sunrise the aircraft must climb from an altitude of approximately 1000 m to a cruising altitude to 8500 m. In order to reduce energy consumption, a fully charged UAV descends to a low-altitude region in the evening to take advantage of lower air resistance and will continue to fly. The ideal charge-discharge process under the flight height profile is shown in Figure 13.

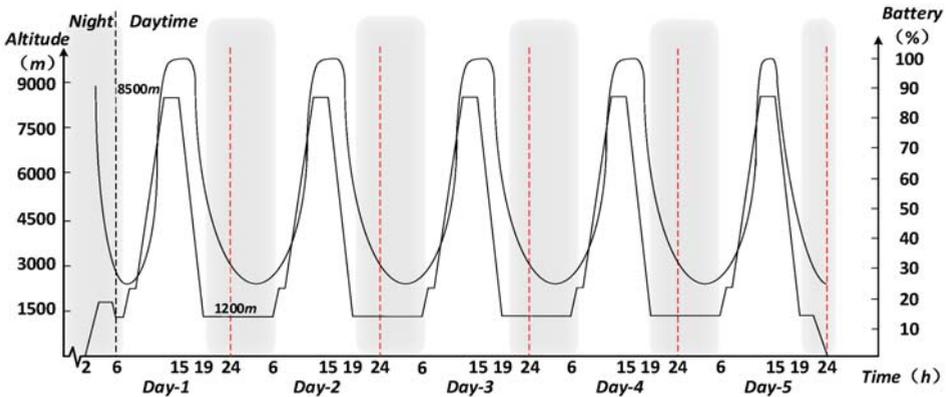


Figure 13. Correspondence between flight height profile of UAV and charge-discharge cycle.

During the process, the ambient temperature will change cyclically with flight height, and the structure with the temperature degradation characteristic will experience significant state change. A simplified temperature stress profile is given in Figure 14.

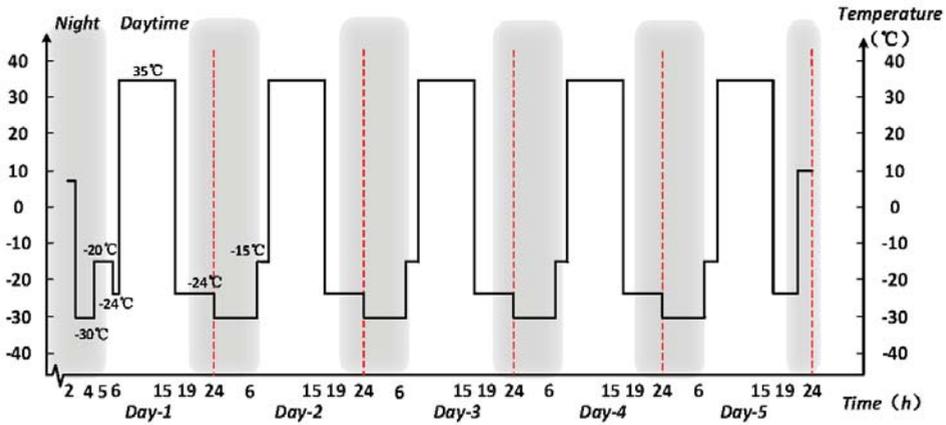


Figure 14. Simplified temperature profile of solar-powered UAV flight.

The lifetime prediction model considering the temperature stress is established by combining the sensor information of the 37 devices, and the state of the cruise mission in the next 35 d is predicted according to the above temperature profile. Figure 15 shows the state probability distribution of the solar-powered unmanned system and its data link, ground subsystem, and airborne subsystem.

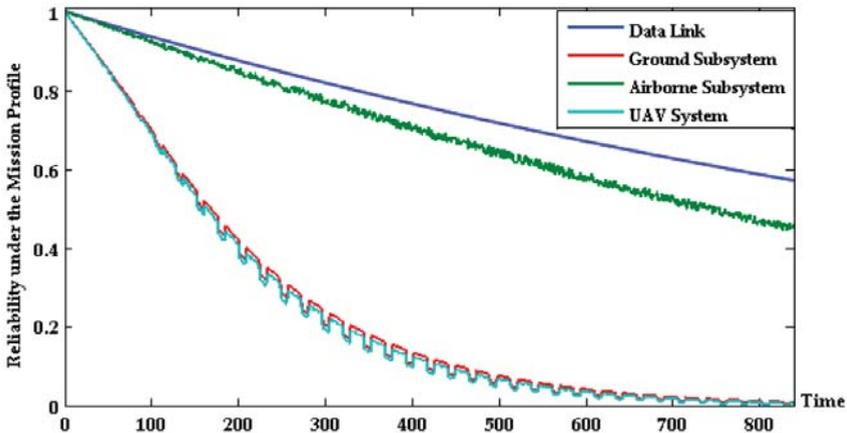


Figure 15. Probability prediction curve of unmanned system state.

The continual decreasing trend reflects the degradation of operation state of the (sub-) system with time and environmental conditions; the UAV system and airborne subsystem show cyclically violent state changes after flying 100 h approximately. This is because the temperature degradation characteristics of the airborne equipment are subject to periodic ambient temperature stress.

5.2. Analysis Based on the Prediction Results

(1) Task risk analysis

In order to make full use of the predicted information to identify the health of the UAV system, a task risk analysis was carried out on the system based on health status. Based on fuzzy

theory, the probabilistic predictive values for each time are divided into four healthy states, such as “healthy”, “minor fault”, “catastrophic fault”, and “close to failure” with the trapezoidal fuzzy number. The probability distribution curves of four states are shown as dotted lines in Figure 16. The risk of each state can be weighted to achieve the risk assessment of system at all times (as shown in Figure 16).

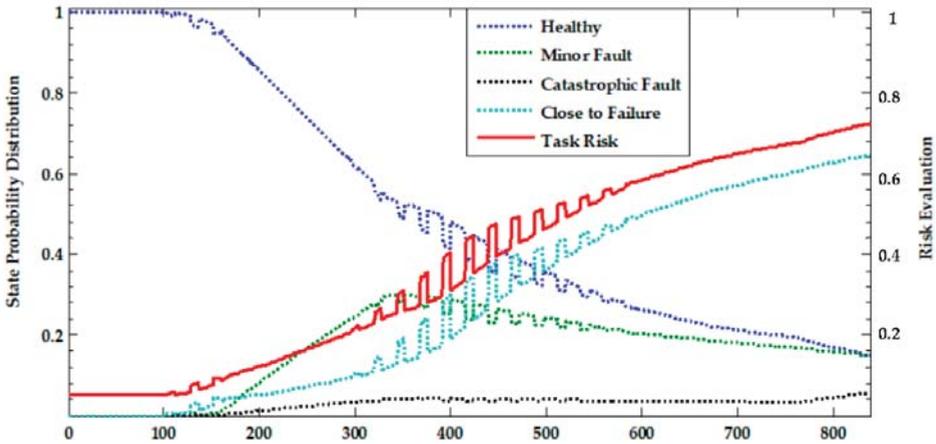


Figure 16. Probability distribution and risk prediction of each state of unmanned system.

It can be seen from Figure 16 that the UAV system can work normally in the first 438 h, but frequent abnormal jumps between “healthy” and “close to failure” occurred between 439 and 496h, and the worst health state was maintained after 497 h, which leads to a continuous increase in systemic risk, especially after 512 h when the task risk has exceeded 0.5. Continuing to perform the task poses a considerable probability of failure and safety risks for personnel and equipment. Additionally, at the time of 439 h when state mutation firstly occurs, the reliability of each device is predicted and shown in Figure 17.

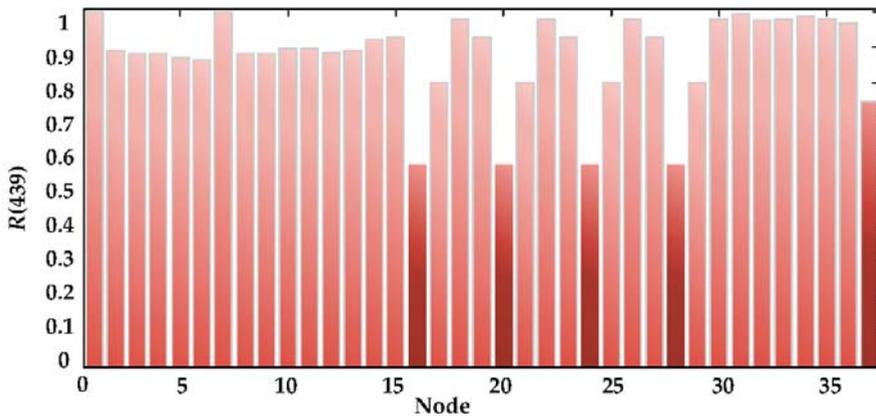


Figure 17. Reliability prediction of each piece of equipment at 439 h.

It can be seen from the figure that the probability that the lithium battery and the data link (nodes 16, 20, 24, 28, and 37) can work normally is relatively low when the system state begins to appear abnormally, which may be the fundamental factor causing the deterioration of the system state.

(2) Sensitivity analysis of nodes

Sensitivity can be used to reflect the sensitivity of system output to the amount of input change so that the weak parts of the system can be identified. Due to the conditional independence assumption in BN, nodes at the same level are independent of each other, and thus the sensitivity of each node to the system is defined as:

$$\Theta_{i,S} = \frac{\partial R_s}{\partial R_{X_i}} = \frac{\Delta R_s}{\Delta R_{X_i}} \quad (13)$$

In the BN model of the UAV system, the sensitivity of each subsystem is shown in Figure 18.

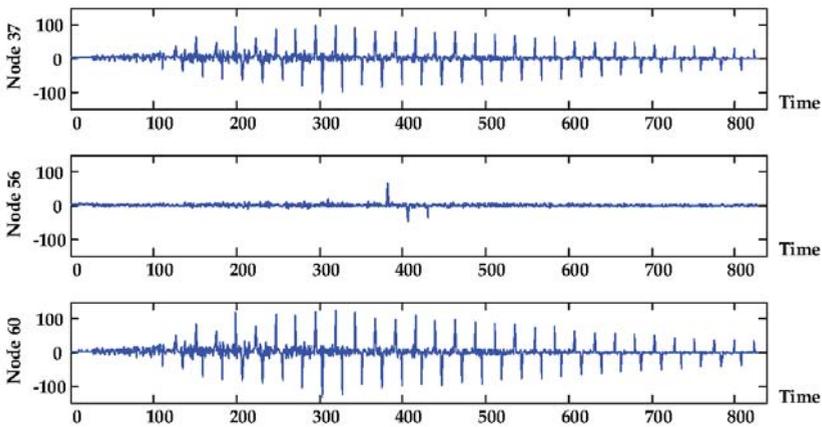


Figure 18. System sensitivity to three subsystems.

According to the figure, the system is more sensitive to the state change of node 60 (airborne subsystem). Further analysis of the airborne subsystem shows that the sensitivity of nodes 16, 20, 24, and 28 can reach 14 orders of magnitude at a certain local time (the figures of which are not shown in this paper). In addition, node 37 (data link) is both a subsystem and root node whose sensitivity is at a high level. As a result, the airborne subsystem and data link subsystem will have a significant impact on the system state.

(3) Recommendations for reliability design and maintenance

In reliability design, sensitivity analysis can be used to improve the inherent reliability of critical equipment by design improvement. Meanwhile, close attention should be paid to redundancy design within the allowable range of aircraft load. This will create significant improvements in the reliability, availability, and longevity of UAV systems. From the point of view of maintenance and repair, the maintenance strategy $T(C, A^*, P_{risk}|t)$ related to maintenance cost $C(t)$, instantaneous availability $A^*(s)$ and work risk $P_{risk}(t)$ can be formulated according to the given lifetime index. Combined with system mission risk and the average cost of preventive and post-maintenance, a maintenance cost-time model can be established. Maintenance or replacement activities are carried out at higher cost-efficiency-ratio point in accordance with the priority of the equipment status and sensitivity.

6. Conclusions

In this paper we propose a modeling method for lifetime prediction of systems from integrated multi-sensor information at device level based on a BN in order to conduct system-level lifetime prediction for complex, multi-sensor systems. The simulation results show that the method can effectively solve the complex logical association of data and the quantitative description of the uncertainty in the multi-level system, and that it has good integration ability and compatibility with various prediction models of devices. At the same time, the application research of a multi-level solar-powered UAV system with multi-device information is carried out, and the task risk and node sensitivity analysis are conducted according to the prediction results of system lifetime based on the temperature profile. The research shows that the system prediction method from the integrated multi-sensor lifetime based on a BN has the following advantages:

- (1) By making full use of multi-sensor information, data association, and quantitative expression, integration and lifetime prediction can be achieved on multi-level, complex, dynamic, multi-source logic data.
- (2) The diversified life expectancy information of the model output is based on different aspects and complement each other, which provides comprehensive data support and theoretical guidance for design improvements and maintenance from system view and in the entire lifecycle process that considers a trade-off of economics, technology, risk, and effectiveness.

However, simulation and application studies show that the proposed method has a high dependence on the prediction model at the device level. The more accurate lifetime prediction for a complex multi-level system is based on accurate device-level prediction information, which imposes higher requirements on information acquisition, processing, and analysis from multiple sensors. Therefore, there might be limitations in practice by only considering data from sensors. Future work on lifetime prediction methods based on a BN should focus on the study and understanding of prediction accuracy, and appropriately take into account other sources and levels data like experimental data and reasonable expert knowledge and so on to conduct integration. In addition, the prediction dimension can be extended from two states to multiple states, and the intermediate process of various mechanisms can be studied in depth by considering the intermediate state to obtain more abundant lifetime prediction information.

Acknowledgments: This work was supported by the Aero-Science Fund (Grant No. 2015ZD51044).

Author Contributions: Jingbin Wang and Lizhi Wang proposed the idea of the research, conceived and designed the simulation experiments; Xiaohong Wang and Lizhi Wang conceived and provided the application case; Jingbin Wang performed the experiments, analyzed the data and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ge, Q.; Wei, Z.; Cheng, T.; Chen, S.; Wang, X. Flexible Fusion Structure-Based Performance Optimization Learning for Multisensor Target Tracking. *Sensors* **2017**, *17*, 1045. [CrossRef] [PubMed]
2. Xu, H.; Liu, J.; Hu, H.; Zhang, Y. Wearable Sensor-Based Human Activity Recognition Method with Multi-Features Extracted from Hilbert-Huang Transform. *Sensors* **2016**, *16*, 2048. [CrossRef] [PubMed]
3. Li, Z.; Feng, L.; Yang, A. Fusion Based on Visible Light Positioning and Inertial Navigation Using Extended Kalman Filters. *Sensors* **2017**, *17*, 1093.
4. Rodger, J.A. Toward reducing failure risk in an integrated vehicle health maintenance system: A fuzzy multi-sensor data fusion Kalman filter approach for IVHMS. *Expert Syst. Appl.* **2012**, *39*, 9821–9836. [CrossRef]
5. Jing, L.; Wang, T.; Zhao, M.; Wang, P. An Adaptive Multi-Sensor Data Fusion Method Based on Deep Convolutional Neural Networks for Fault Diagnosis of Planetary Gearbox. *Sensors* **2017**, *17*, 414. [CrossRef] [PubMed]

6. Hsu, Y.L.; Chou, P.H.; Chang, H.C.; Lin, S.L.; Yang, S.C.; Su, H.Y.; Chang, C.C.; Cheng, Y.S.; Kuo, Y.C. Design and Implementation of a Smart Home System Using Multisensor Data Fusion Technology. *Sensors* **2017**, *17*, 1631. [CrossRef] [PubMed]
7. Whitmore, G.A.; Schenkelberg, F. Modelling accelerated degradation data using Wiener diffusion with a time scale transformation. *Lifetime Data Anal.* **1997**, *3*, 27–45. [CrossRef] [PubMed]
8. Bagdonavicius, V.; Nikulin, M.S. Estimation in degradation models with explanatory variables. *Lifetime Data Anal.* **2001**, *7*, 85–103. [CrossRef] [PubMed]
9. Wan, Y.; Huang, H.L.; Das, D.; Pecht, M. Thermal reliability prediction and analysis for high-density electronic systems based on the Markov process. *Microelectron. Reliab.* **2016**, *56*, 182–188. [CrossRef]
10. Ali, J.B.; Chebel-Morello, B.; Saidi, L.; Malinowski, S.; Fnaiech, F. Accurate Bearing remaining useful life prediction based on Weibull distribution and artificial neural network. *Mech. Syst. Sig. Process.* **2015**, *56*, 150–172.
11. Alghassi, A.; Perinpanayagam, S.; Samie, M. Stochastic RUL Calculation Enhanced With TDNN-Based IGBT Failure Modeling. *IEEE Trans. Reliab.* **2016**, *65*, 558–573. [CrossRef]
12. Son, J.; Zhou, S.; Sankavaram, C.; Sankavaram, C.; Du, X.; Zhang, Y. Remaining useful life prediction based on noisy condition monitoring signals using constrained Kalman filter. *Reliab. Eng. Syst. Saf.* **2016**, *152*, 38–50. [CrossRef]
13. Li, K.; Wu, J.; Zhang, Q.; Su, L.; Chen, P. New Particle Filter Based on GA for Equipment Remaining Useful Life Prediction. *Sensors* **2017**, *17*, 696. [CrossRef] [PubMed]
14. Qian, Y.N.; Yan, R.Q.; Gao, R.X. A multi-time scale approach to remaining useful life prediction in rolling bearing. *Mech. Syst. Sig. Process.* **2017**, *83*, 549–567. [CrossRef]
15. Nguyen, K.A.; Do, P.; Grall, A. Multi-level predictive maintenance for multi-component systems. *Reliab. Eng. Syst. Saf.* **2015**, *144*, 83–94. [CrossRef]
16. Zheng, X.; Fang, H. An integrated unscented kalman filter and relevance vector regression approach for lithium-ion battery remaining useful life and short-term capacity prediction. *Reliab. Eng. Syst. Saf.* **2015**, *144*, 74–82. [CrossRef]
17. Khorasgani, H.; Biswas, G.; Sankararaman, S. Methodologies for system-level remaining useful life prediction. *Reliab. Eng. Syst. Saf.* **2016**, *154*, 8–18. [CrossRef]
18. Coolen, F.P.A.; Coolen-Maturi, T. The structure function for system reliability as predictive (imprecise) probability. *Reliab. Eng. Syst. Saf.* **2016**, *154*, 180–187. [CrossRef]
19. Jackson, C.; Mosleh, A. Bayesian inference with overlapping data: Reliability estimation of multi-state on-demand continuous life metric systems with uncertain evidence. *Reliab. Eng. Syst. Saf.* **2016**, *145*, 124–135. [CrossRef]
20. George-Williams, H.; Patelli, E. A hybrid load flow and event driven simulation approach to multi-state system reliability evaluation. *Reliab. Eng. Syst. Saf.* **2016**, *152*, 351–367. [CrossRef]
21. Meineri, E.; Dahlberg, C.J.; Hylander, K. Using Gaussian Bayesian Networks to disentangle direct and indirect associations between landscape physiography, environmental variables and species distribution. *Ecol. Modell.* **2015**, *313*, 127–136. [CrossRef]
22. Pearl, J. Fusion, propagation, and structuring in belief networks. *Artif. Intell.* **1986**, *29*, 241–288. [CrossRef]
23. Weber, P.; Medina-Oliva, G.; Simon, C.; Iung, B. Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas. *Eng. Appl. Artif. Intell.* **2012**, *25*, 671–682. [CrossRef]
24. Tien, I.; Kiureghian, A.D. Algorithms for Bayesian network modeling and reliability assessment of infrastructure systems. *Reliab. Eng. Syst. Saf.* **2016**, *156*, 134–147. [CrossRef]
25. Price, L.C.; Peiris, H.V.; Frazer, J.; Easther, R. Designing and testing inflationary models with Bayesian networks. *J. Cosmol. Astropart. Phys.* **2016**, *2016*, 049. [CrossRef]
26. Garvey, M.D.; Carnovale, S.; Yenyiurt, S. An analytical framework for supply network risk propagation: A Bayesian network approach. *Eur. J. Oper. Res.* **2015**, *243*, 618–627. [CrossRef]
27. Cai, B.; Liu, H.; Xie, M. A real-time fault diagnosis methodology of complex systems using object-oriented Bayesian networks. *Mech. Syst. Sig. Process.* **2016**, *80*, 31–44. [CrossRef]
28. Cai, B.; Liu, Y.; Fan, Q.; Zhang, Y.; Liu, Z.; Yu, S.; Ji, R. Multi-source information fusion based fault diagnosis of ground-source heat pump using Bayesian network. *Appl. Energy* **2014**, *114*, 1–9. [CrossRef]
29. Khakzad, N. Application of dynamic Bayesian network to risk analysis of domino effects in chemical infrastructures. *Reliab. Eng. Syst. Saf.* **2015**, *138*, 263–272. [CrossRef]

30. Tang, D.; Makis, V.; Jafari, L.; Yu, J. Optimal maintenance policy and residual life estimation for a slowly degrading system subject to condition monitoring. *Reliab. Eng. Syst. Saf.* **2015**, *134*, 198–207. [CrossRef]
31. Cai, B.; Zhao, Y.; Liu, H.; Xie, M. A Data-Driven Fault Diagnosis Methodology in Three-Phase Inverters for PMSM Drive Systems. *IEEE Trans. Power Electron.* **2017**, *32*, 5590–5600. [CrossRef]
32. Yontay, P.; Pan, R. A computational Bayesian approach to dependency assessment in system reliability. *Reliab. Eng. Syst. Saf.* **2016**, *152*, 104–114. [CrossRef]
33. Hu, J.; Zhang, L.; Ma, L.; Liang, W. An integrated safety prognosis model for complex system based on dynamic Bayesian network and ant colony algorithm. *Expert Syst. Appl.* **2011**, *38*, 1431–1446. [CrossRef]
34. Li, N.; Feng, X.; Jimenez, R. Predicting rock burst hazard with incomplete data using Bayesian networks. *Tunn. Undergr. Space Technol.* **2017**, *61*, 61–70. [CrossRef]
35. Kabir, G.; Demissie, G.; Sadiq, R.; Tesfamariam, S. Integrating failure prediction models for water mains: Bayesian belief network based data fusion. *Knowl.-Based Syst.* **2015**, *85*, 159–169. [CrossRef]
36. Murphy, K. Bayesian Network Toolbox (BNT). Available online: <http://www.cs.ubc.ca/murphyk/Software/BNT/bnt.html> (accessed on 11 August 2017).



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

A Robust Vehicle Localization Approach Based on GNSS/IMU/DMI/LiDAR Sensor Fusion for Autonomous Vehicles

Xiaoli Meng ¹, Heng Wang ^{2,*} and Bingbing Liu ¹

¹ Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore; xiaoli.meng09@gmail.com (X.M.); bliu@i2r.a-star.edu.sg (B.L.)

² College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

* Correspondence: wangheng@ise.neu.edu.cn; Tel.: +86-024-83681939

Received: 25 August 2017; Accepted: 15 September 2017; Published: 18 September 2017

Abstract: Precise and robust localization in a large-scale outdoor environment is essential for an autonomous vehicle. In order to improve the performance of the fusion of GNSS (Global Navigation Satellite System)/IMU (Inertial Measurement Unit)/DMI (Distance-Measuring Instruments), a multi-constraint fault detection approach is proposed to smooth the vehicle locations in spite of GNSS jumps. Furthermore, the lateral localization error is compensated by the point cloud-based lateral localization method proposed in this paper. Experiment results have verified the algorithms proposed in this paper, which shows that the algorithms proposed in this paper are capable of providing precise and robust vehicle localization.

Keywords: sensor fusion; Unscented Kalman Filter (UKF); vehicle localization

1. Introduction

Automated driving techniques are widely admitted as a promising and challenging way to avoid road crashes and improve traffic conditions [1]. To make an autonomous vehicle drive safely in urban environments, the vehicle needs to know its exact position and orientation. Thus, localization plays a key role in autonomous vehicle applications. Using the popular GNSS (Global Navigation Satellite System) for localization requires a GNSS receiver with an unobstructed line of sight to four or more GNSS satellites. However, even with a high-end GNSS-based system, a vehicle's location may jump up to a few meters as different satellites go in and out of view or obstructions in the environments create multi-path interferences. INS (Inertial Navigation System) is complementary to GNSS as it does not rely on external information sources, which can be blocked or disturbed. INS can provide complete navigation information such as position, velocity and attitude by integrating the accelerometer and gyroscope readings over time. However, as the inertial sensors of an INS are subject to drifts, navigation systems based on stand-alone INS suffer a rapid degradation of position over time. This is particularly true when a low-cost IMU (Inertial Measurement Unit) is employed. In the dead-reckoning integration scheme, wheel encoders are always introduced to slow the rate of growth of IMU integration errors, but are subject to errors due to wheel slip. However, a robust localization solution can be achieved by blending GNSS, INS and DMI (Distance Measuring Instruments) techniques in a way that utilizes the strengths of each individual system and mitigates their weaknesses.

To fuse GNSS, INS and DMI, Extended Kalman Filtering (EKF) is a popular sensor fusion method [2,3], where nonlinear systems are linearized and approximated around current state estimates. However, in the EKF, high-order terms are neglected, which are necessary for some situations. The Particle Filter (PF) [4,5] is another useful method, but the main drawback of this filter is its computational requirement, which makes it not very suitable for real-time applications. In [6],

an INS/GPS sensor fusion scheme based on the State-Dependent Riccati Equation (SDRE) nonlinear filtering method is proposed for Unmanned Aerial Vehicles (UAV), which is widely used in the optimal nonlinear control and filtering literature. Although this method provides an alternative INS/GPS filtering scheme, the real-time performance and application to autonomous vehicles in urban environments are still uncertain. In [7,8], low cost sensors such as cameras are fused with GNSS/INS to improve the localization accuracy of GNSS in dense urban areas where obstacles block satellite signals; however, the improvement of accuracy is limited due to the difficulties in some feature recognition tasks and the calculating of depth information using cameras. Recently, the Unscented Kalman Filter (UKF) has been used for localization based on GPS/INS sensor fusion [9–11] due to the ability to remove the messy Jacobian matrix computation and keep at least a second-order nonlinear function approximation. Although UKF has been proven to be a promising method for GPS/INS fusion, the accuracy and reliability performance still need to be improved for autonomous vehicles under urban environments.

On the other hand, the localization approach based solely on GNSS/IMU/DMI cannot always guarantee a precise location solution due to the existence of the blocking of satellite signals by obstacles (buildings and trees, etc.) and the cumulative errors of IMU and DMI sensors. In order to provide precise localization for positioning an autonomous vehicle reliably, we need to explore other useful information to position an autonomous vehicle. In the urban environment, curbs and lane markings comprise two kinds of useful information for improving the results of GPS/INS/DMI fusion. For example, in [12–15], cameras are fused with other sensors such as GPS and IMU to improve the lateral accuracy by detecting lane markings. However, the detection of lane marking needs to face the challenges of different lighting, poor lane markings, etc. On the other hand, 3D point clouds generated by a 3D LiDAR scanner provide more reliable performance for curb detection, which can also be used to improve the lateral localization accuracy [16–20]. Furthermore, in [21], an eigenvector technique was used to find a line segment corresponding to edges of roads. In [22], a Hough transform was used to find the best fit line to the surface on the road, and points corresponding to the best fit line were used as curb points of the road. In [23], curbs are extracted by using a 1D laser scanner.

In this paper, to further improve the accuracy and reliability of localization for autonomous vehicles in urban environments, we firstly propose a fault-detection-based loosely-coupled GNSS/IMU/DMI localization solution, which can improve the performance of the traditional UKF-based method; then, we correct the lateral localization errors based on curb detection results using a multi-layer LiDAR. The flowchart of overall localization method is summarized in Figure 1.

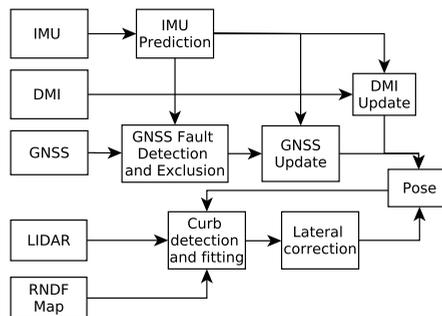


Figure 1. The flowchart of the proposed method. DMI, Distance-Measuring Instrument; RNDF, Route Network Definition File.

The main contribution of this paper is summarized as follows. Firstly, through combining the fault-detection method with the UKF-based GNSS/IMU/DMI fusion algorithm, the localization

accuracy of autonomous vehicles is improved greatly; Secondly, a point cloud-based curb detection and fitting method is proposed to improve the lateral accuracy of the autonomous vehicle further, where the RANSAC algorithm is utilized. The rest of the paper is organized as follows: Section 2 presents the proposed UKF-based localization approach: firstly, the modeling including the process model and the measurement model is introduced, followed by the implementation of the UKF. Details about curb-based lateral localization are provided in Section 3. Section 4 presents the experimental results. Finally, we conclude the paper in Section 5.

2. UKF-Based Localization Approach

For a vehicle localization system, four coordinate systems are defined:

- Earth-Centered-Earth-Fixed (ECEF) coordinate system (E): It has an origin at the center of the Earth. The positive Z-axis goes out the Earth's north pole; the X-axis is along the prime meridian; and the Y-axis completes the right-handed system;
- Global coordinate system (G): The North-East-Down (NED) coordinate system is defined as G with the X-axis pointing north, the Y-axis pointing east and the Z-axis pointing down to construct a right-handed coordinate system;
- Body coordinate system (B): The coordinate system of the vehicle with the X-axis pointing forwards, the Y-axis pointing left and the Z-axis pointing up;
- Sensor coordinate system (S): the three orthogonal axes of the mounted sensors. We assume that S coincides with B after sensor to body alignment calibration [24].

One should note that each sensor defines its own coordinate system. We need to note the difference between the origins for accurate localizations.

We describe the states of the filtering system with the following vector:

$$\mathbf{x} = [\mathbf{p}, \mathbf{v}, \mathbf{q}, \mathbf{b}, \mathbf{d}]_{16 \times 1}^T \quad (1)$$

where \mathbf{p} and \mathbf{v} are the position and velocity of the vehicle within the global frame G, respectively. \mathbf{q} is a unit quaternion that represents the rotation from the body frame B to the global frame G. A unit quaternion consists of a vector part $\mathbf{e} = (q_1, q_2, q_3)^T \in \mathbb{R}^3$ and a scalar part $q_4 \in \mathbb{R}$ [25]:

$$\mathbf{q} = [\mathbf{e}^T, q_4]^T = [q_1, q_2, q_3, q_4]^T$$

and its norm equals one, that is:

$$\|\mathbf{q}\| = 1.$$

As accelerometers and gyroscopes have biases, which can be modeled as random walk processes, we have two additional vectors \mathbf{b} and \mathbf{d} in the state vector to represent their biases, respectively. Both variables are given within body frame B.

2.1. Process Model

The process model governs the dynamic relationship between the states of two successive time steps, which can be described by:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \mathbf{w}_{t-1} \quad (2)$$

where \mathbf{x}_t is the predicted state after time period δ based on the last known state vector \mathbf{x}_{t-1} , \mathbf{u}_{t-1} is the input to the state space models and \mathbf{w}_{t-1} is the process noise. In this study, we use the following process model:

$$\begin{aligned} \mathbf{x}_t &= f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \mathbf{w}_{t-1} \\ &= \begin{bmatrix} \mathbf{p}_{t-1} + \mathbf{v}_{t-1}\delta \\ \mathbf{v}_{t-1} + \mathbf{C}(\mathbf{q}_{t-1})(\mathbf{y}_{a,t-1} - \mathbf{b}_{t-1})\delta - \mathbf{g}\delta \\ \exp\left(\frac{1}{2}\Omega[\mathbf{y}_{\omega,t-1} - \mathbf{d}_{t-1}]\delta\right) \mathbf{q}_{t-1} \\ (1 - 1/\tau_a)\mathbf{b}_{t-1} \\ (1 - 1/\tau_\omega)\mathbf{d}_{t-1} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{p,t-1} \\ \mathbf{w}_{v,t-1} \\ \mathbf{w}_{q,t-1} \\ \mathbf{w}_{b,t-1} \\ \mathbf{w}_{d,t-1} \end{bmatrix} \end{aligned} \quad (3)$$

where $\mathbf{u}_{t-1} = [\mathbf{y}_{a,t-1}, \mathbf{y}_{\omega,t-1}]$ is the measurement vector from the accelerometer and gyroscope at time step $t-1$, $\mathbf{C}(\mathbf{q}_{t-1})$ is the corresponding rotation matrix of the quaternion \mathbf{q}_{t-1} [26]:

$$\mathbf{C}(\mathbf{q}_{t-1}) = \left(q_{4,t-1}^2 - \mathbf{e}_{t-1}^T \mathbf{e}_{t-1} \right) \mathbf{I}_3 + 2\mathbf{e}_{t-1} \mathbf{e}_{t-1}^T - 2q_{4,t-1} [\mathbf{e}_{t-1}]_\times$$

representing the transformation from the body frame B to the global frame G, \mathbf{g} is the gravitational acceleration vector and $\Omega[\boldsymbol{\omega}]$ is a 4×4 skew symmetric matrix, as in:

$$\Omega[\boldsymbol{\omega}] = \begin{bmatrix} -[\boldsymbol{\omega}]_\times & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}$$

where $[\boldsymbol{\omega}]_\times$ is defined by

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega^Z & \omega^Y \\ \omega^Z & 0 & -\omega^X \\ -\omega^Y & \omega^X & 0 \end{bmatrix}$$

where $[\omega^X, \omega^Y, \omega^Z]$ are the three elements of angular rates on the X-, Y- and Z-axis, respectively, and τ_a and τ_ω are time constants. Process noise is added by the noise vector $\mathbf{w}_{t-1} = [\mathbf{w}_{p,t-1}, \mathbf{w}_{v,t-1}, \mathbf{w}_{q,t-1}, \mathbf{w}_{b,t-1}, \mathbf{w}_{d,t-1}]^T$. Each noise item is modeled as a zero mean Gaussian noise with covariance matrix $\mathbf{Q}_p, \mathbf{Q}_v, \mathbf{Q}_q, \mathbf{Q}_b, \mathbf{Q}_d$, respectively. The noise items are assumed to be uncorrelated with each other; thus, the process noise covariance matrix has the following expressions:

$$\mathbf{Q} = \text{diag}[\mathbf{Q}_p, \mathbf{Q}_v, \mathbf{Q}_q, \mathbf{Q}_b, \mathbf{Q}_d]. \quad (4)$$

2.2. Measurement Model

The measurement model governs the relationship between the state vector and sensor measurements, which is:

$$\mathbf{y}_t = g(\mathbf{x}_t) + \mathbf{n}_t \quad (5)$$

In this paper, measurements from the GNSS receiver and encoder are introduced to bound the errors in estimates of the vehicle position/velocity and attitude. Asynchronous updates are performed within the UKF as measurements become available from the wheel encoder and GNSS receiver. For wheel encoder measurements, a filter update is calculated from the measured speed of the vehicle. For GNSS measurements, a filter update is calculated from the location of the Trimble.

2.2.1. Measurement Model of GNSS

The GNSS receiver could be a differential GNSS receiver or RTK receiver, which measures data at a relatively low frequency (the measurement update rate is up to 20 Hz). When a measurement from the GNSS receiver is available, the GNSS measurement model is given by:

$$\mathbf{y}_p = \mathbf{p} + \mathbf{n}_p \quad (6)$$

where \mathbf{n}_p is the GNSS measurement noise modeled as a Gaussian noise, $N(0, \Sigma_{GNSS})$, $\mathbf{y}_p = \mathbf{C}_E^G [p^X, p^Y, p^Z]$, and $[p^X, p^Y, p^Z]$ is the position on the X-, Y- and Z-axis in the ECEF frame, calculated by the following equations, respectively:

$$\begin{aligned} p^X &= (N + h) \cos \lambda \cos \phi \\ p^Y &= (N + h) \cos \lambda \sin \phi \\ p^Z &= [N(1 - e^2) + h] \sin \lambda \end{aligned}$$

where $[\lambda, \phi, h]$ are the latitude, longitude and altitude provided by GNSS fixes. The parameters used above are defined as follows: $N = a / \sqrt{1 - e^2} \sin^2 \lambda$ is the length from the center of the Earth to the surface; $e = \sqrt{1 - b^2 / a^2}$ is the Earth eccentricity; and $a = 6,378,137$ (m), $b = 6,356,752.3142$ (m) are the Earth ellipsoid semi-major and semi-minor axes, respectively. The transition matrix from the ECEF frame to the global frame G , is denoted by \mathbf{C}_E^G as

$$\mathbf{C}_E^G = \begin{bmatrix} -\sin \lambda \cos \phi & -\sin \lambda \sin \phi & \cos \lambda \\ -\sin \phi & -\cos \phi & 0 \\ -\cos \lambda \cos \phi & -\cos \lambda \sin \phi & -\sin \lambda \end{bmatrix}.$$

The covariance matrix of the GNSS measurement noise is $\mathbf{R} = \Sigma_{GNSS}$.

2.2.2. Measurement Model of DMI

The measurement model for the encoder is modeled as:

$$\mathbf{y}_v = \mathbf{q}^{-1} \otimes \mathbf{v} \otimes \mathbf{q} + \mathbf{n}_v \tag{7}$$

where $\mathbf{y}_v = [v, 0, 0]$ is the velocity of the vehicle in the body frame, v is the wheel speed measurement from the encoder and \otimes represents the quaternion multiplication [25]. \mathbf{n}_v is the encoder measurement noise modeled as a Gaussian noise, $N(0, \Sigma_{Encoder})$, and the covariance matrix of the encoder measurement noise is $\mathbf{R} = \Sigma_{encoder}$.

2.3. Implementation of UKF

In process Model (3), due to the nonlinearity in the quaternion and velocity state functions, unscented transform-based approximation to the optimal filtering solution can be derived by executing two steps of time update and measurement update in turn. In the execution of the two steps, unscented transform is always carried out first to form the sigma points of the state vector.

2.3.1. Time Update

At time step t , sigma points need to be calculated first and followed by performing the time update using time update equations.

- Calculate the sigma points:

$$\tilde{\mathbf{x}}_{t-1} = \left[\mathbf{x}_{t-1} \quad \mathbf{x}_{t-1} \pm \sqrt{(n + \kappa) \mathbf{P}_{t-1}} \right] \tag{8}$$

where $\tilde{\mathbf{x}}_{t-1}$ are the sigma points of state vector \mathbf{x} at previous time step $t - 1$, n is the dimension of the state vector \mathbf{x} , $\kappa = \alpha^2 (n + \gamma) - n$. α determines the spread of the sigma points and γ is a secondary scaling parameter, which is usually set to one. One should note that the initial condition $\mathbf{x}_0 \sim N(\mathbf{x}_0, \mathbf{P}_0)$ should be known.

- Time update process:

$$\bar{\chi}_{i,t|t-1} = f(\chi_{i-1}, \mathbf{u}_{i-1}) \tag{9}$$

$$\mathbf{x}_t^- = \sum_{i=0}^{2n} \mathbf{W}_i^m \bar{\chi}_{i,t|t-1} \tag{10}$$

$$\mathbf{P}_t^- = \sum_{i=0}^{2n} \mathbf{W}_i^c (\bar{\chi}_{i,t|t-1} - \mathbf{x}_t^-) (\bar{\chi}_{i,t|t-1} - \mathbf{x}_t^-)^T + \mathbf{Q} \tag{11}$$

where \mathbf{x}_t^- and \mathbf{P}_t^- are the predicted mean and covariance, respectively, and \mathbf{W}_i^m and \mathbf{W}_i^c are the weights of mean and covariance, which are associated with the i -th point, given by [27]:

$$\begin{aligned} \mathbf{W}_0^m &= \frac{\kappa}{n + \kappa} \\ \mathbf{W}_0^c &= \frac{\kappa}{n + \kappa} + (1 - \alpha^2 + \beta) \\ \mathbf{W}_i^m &= \mathbf{W}_i^c = \frac{1}{2(n + \kappa)}, i = 1, 2, \dots, 2n \end{aligned}$$

where β is a parameter used to incorporate any prior knowledge about the distribution of state \mathbf{x} (for Gaussian distributions, $\beta = 2$ is optimal).

2.3.2. Measurement Update of GNSS

When fix measurement $\mathbf{y} = \mathbf{y}_p$ is available, we can update the nearest state prediction \mathbf{x}_t^- and covariance matrix \mathbf{P}_t^- using the following equations. At first, sigma points need to be calculated and then, measurement update is performed using measurement update equations.

- Calculate the sigma points:

$$\bar{\chi}_t = \left[\mathbf{x}_t^- \quad \mathbf{x}_t^- \pm \sqrt{(n + \kappa) \mathbf{P}_t^-} \right] \tag{12}$$

where \mathbf{x}_t^- and \mathbf{P}_t^- are the predicted mean and covariance from time update at time t , respectively.

- Perform measurement update:

$$\bar{\mathbf{Y}}_t = g(\bar{\chi}_t) \tag{13}$$

$$\bar{\mathbf{y}}_t = \sum_{i=0}^{2n} \mathbf{W}_i^m \bar{\mathbf{Y}}_{i,t} \tag{14}$$

$$\mathbf{P}_{y_t} = \sum_{i=0}^{2n} \mathbf{W}_i^c (\bar{\mathbf{Y}}_{i,t} - \bar{\mathbf{y}}_t) (\bar{\mathbf{Y}}_{i,t} - \bar{\mathbf{y}}_t)^T + \mathbf{R} \tag{15}$$

$$\mathbf{P}_{x_t y_t} = \sum_{i=0}^{2n} \mathbf{W}_i^c (\bar{\chi}_{i,t} - \mathbf{x}_t^-) (\bar{\mathbf{Y}}_{i,t} - \bar{\mathbf{y}}_t)^T \tag{16}$$

$$\mathbf{K}_t = \mathbf{P}_{x_t y_t} (\mathbf{P}_{y_t})^{-1} \tag{17}$$

$$\mathbf{v}_t = \mathbf{y}_{p,t} - \bar{\mathbf{y}}_t \tag{18}$$

$$\mathbf{x}_t = \mathbf{x}_t^- + \mathbf{K}_t \mathbf{v}_t \tag{19}$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t \mathbf{P}_{y_t} \mathbf{K}_t^T \tag{20}$$

where $\bar{\mathbf{Y}}_t$ are the projected sigma points through the measurement function h , $\bar{\mathbf{y}}_t$ is the predicted measurement produced by the weighted sigma points, \mathbf{P}_{y_t} and $\mathbf{P}_{x_t y_t}$ are the predicted measurement covariance and the state-measurement cross-covariance matrix, respectively, \mathbf{K}_t is the Kalman gain, \mathbf{v}_t is the innovation and \mathbf{x}_t and \mathbf{P}_t are the updated state and covariance at time t , respectively.

2.3.3. Measurement Update of DMI

When wheel speed measurement $\mathbf{y} = \mathbf{y}_v$ is available, we can update the nearest state prediction \mathbf{x}_f^- and covariance matrix \mathbf{P}_f^- using the similar equations with fix measurement update. Similarly, sigma points need to be calculated, and then, measurement update is performed using measurement update equations.

2.4. Automatic Detection the Degradation of GNSS Performance

GNSS suffers multi-path errors if the satellite signals are reflected off one or more surfaces before reaching the receiver antenna. A different set of satellites is utilized for fix determination, which can also alter the GNSS fix. Big gaps occur when GNSS signals become available again after short-term GNSS dropouts due to the presence of trees and buildings. Different checks are adopted to make our pose estimator robust to the aforementioned jumps, which are explained as follows:

Zero-Velocity update: As GNSS receivers due to pseudo-random error cannot output fixed position information when the vehicle stays in a stationary position, if the vehicle is not in motion, which can be well detected from the encoder readings, measurements from the receiver are not used for updates. This will restrict the vehicle to a fixed location.

Number of satellites: If the number of satellites visible to the receiver is four or more, the measurements from the receiver pass the check.

Dilution Of Precision (DOP): If Horizontal DOP (HDOP) or Vertical DOP (VDOP) is larger than a threshold, the measurements will be discarded.

Statistical test: During the GNSS measurements update, if an abrupt jump in the GNSS fix occurs, the correction made by the GNSS measurement update will cause the IMU solutions to incorrectly follow these jumps. The chi-squared test is applied once the innovation \mathbf{v} and the innovation covariance matrix Σ_v are obtained in the measurement update [28], as in:

$$\mathbf{v}^T \Sigma_v^{-1} \mathbf{v} \leq \zeta \quad (21)$$

where the value ζ is usually set to reject the innovations exceeding the 95% threshold. During the GNSS measurement update stage, if (21) holds, then the GNSS fix is accepted, and the measurement update proceeds.

Assessing the new horizontal position reading and subtracting it from the current estimate of position: If the difference is much higher than what it should be when compared to the vehicle speed obtained from the encoder, which is assumed to be an accurate quantity, then the measurements are discarded.

Check the altitude component: If the measurements are much higher than the innovation, the updates will be aborted.

Validity of position change: After the GNSS measurement update, the change in position $\Delta \mathbf{p}$ can be calculated by:

$$\Delta \mathbf{p} = \mathbf{p}_t - \mathbf{p}_{t-1} \quad (22)$$

The change in position is invalid if it satisfies:

$$|\Delta \mathbf{p}| > v(1 + \eta)\Delta t + \epsilon \vee \left(|\Delta \mathbf{p}| > \epsilon \wedge \frac{|\Delta \mathbf{p}|}{|\Delta \mathbf{p}|} \cdot \begin{bmatrix} \cos \psi \\ \sin \psi \end{bmatrix} > \tau \right) \quad (23)$$

where ψ represents the travel heading, calculated from quaternion \mathbf{q} [25]. η , ϵ and τ are three constants representing the anticipated percentage velocity error, allowed position jitter and allowable travel direction error, respectively. When $\Delta \mathbf{p}$ is rejected, a predicted position is calculated based on heading and wheel speed.

The accuracy of the estimated location of the vehicle is about 1–3 cm or 1 m depending on RTK or if the differential mode is operating. The resulting lateral offset will not guarantee safe driving for autonomous vehicles. In the following section, we are going to introduce the lateral localization based on LiDAR measurements to improve the accuracy of the localization results.

3. Correction of Lateral Localization Errors

After we get the localization result from the fusion of GNSS/IMU/encoder, the lateral localization error can be calculated as follows. The detection of curbs using LiDAR (Light Detection And Ranging) provides an accurate lateral distance between the vehicle and curb (denote it by d_1); at the same time, the same distance can also be calculated using the localization result from GNSS/encoder and RNDF (Route Network Definition File) information (denote it by d_2). The difference of d_1 and d_2 , i.e., $d_1 - d_2$, is the lateral localization error, which can readily be corrected.

Assume that the RNDF (Route Network Definition File) information is accurate enough: given the vehicle position provided by the GNSS/IMU/DMI fusion system, a lateral distance from the vehicle to the curb can always be obtained through a simple geometry calculation; see the yellow line segment in Figure 2.

However, as known to all, the result of the GNSS/IMU/DMI fusion system is always affected by drifts due to the signal failure of GNSS caused by the complexity of the urban environment. This drift may cause the vehicle to hit the curb or rush to the next lane, which may cause a serious accident. In Figure 3, the pink line is the curb line that is fitted by the point cloud collected by a 3D LiDAR, whose accuracy is less than 10 cm, and the orange line is the curb line of the RNDF; obviously, there is a large difference between the vehicle to these two lines, which is caused by the drift of the localization result. Thus, a lateral correction is very necessary for the localization of the autonomous vehicle. To accomplish the goal of LiDAR-based lateral error correction, we first detect longitudinal curbs and use them as measurements to estimate the autonomous vehicle's lateral distance, then compare it with the distance calculated from RNDF, e.g., the yellow line as shown in Figure 2. The difference of these two distances is then used to correct the lateral error of the autonomous vehicle.



Figure 2. Lateral distance from the vehicle to the curb.

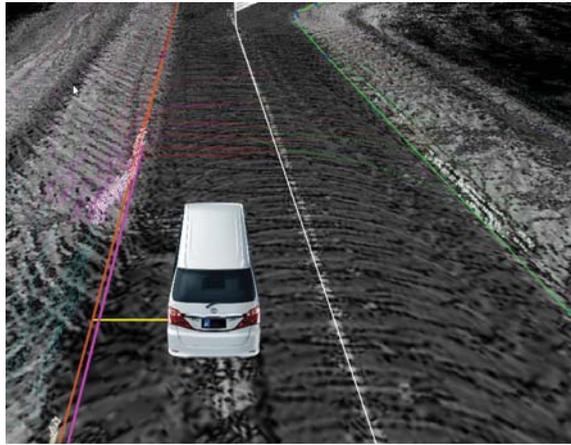


Figure 3. Curb estimated by fitting.

3.1. Curb Detection

3.1.1. Curb Detection Principle

The first step for lateral correction is the curb detection. Figure 4 shows the curb detection principle. Assume that A, B, C, D, E are part of the adjacent points collected by one beam of a 3D LiDAR; the vectors $\vec{AB}, \vec{BC}, \vec{CD}, \vec{DE}$ can be calculated, and B, C can be selected as the curb point since the angle between \vec{BC} and the ground is very large, the same for vector \vec{CD} .

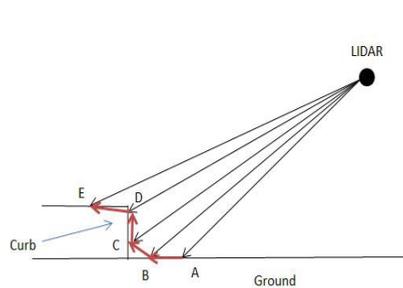


Figure 4. Points selected on the curb. A, B, C, D are 3D points obtained by LiDAR. B and C are selected as points on the curb since the vectors \vec{BC} and \vec{CD} are more “upcast”.

3.1.2. Algorithm for Curb Detection

This subsection provides Algorithm 1 which is used for curb detection.

The following Figure 5 presents an example of the curb detection result where curb points are marked as white points.

Algorithm 1 Framework of curb detection.

Require:

Point clouds collected by a 3D LiDAR;

Ensure:

Step 1: Given input point cloud, select the area of interest;

Step 2: Calculate the vector difference of adjacent points in each beam;

Step 3: Select curb-like points and filter out noises;

Step 4: Separate higher obstacles by comparing the height with a threshold;

return Curb points.

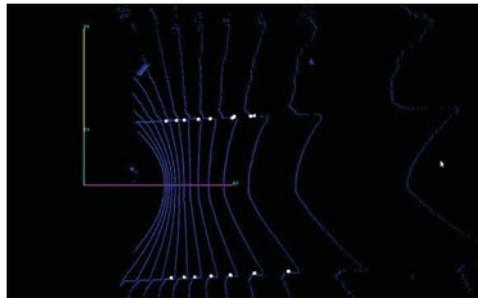


Figure 5. Output to the curb detection module.

3.2. Curb Line Fitting Using RANSAC

After detecting the curb, the next step is to fit the curb line using the RANSAC algorithm, which is an iterative method to estimate the parameters of a curve from detected curb points. The details are presented in the following Algorithm 2.

Denote S as a set of curb points that are separated from higher obstacles, then we have

Algorithm 2 Framework of curb fitting.

Require:

Detected curb points;

Ensure:

Step 1: Randomly select a sample of s curb points from S , and instantiate the model from this subset;

Step 2: Determine the set of curb points S_i that are within a distance threshold t of the model. The set S_i is the consensus set of samples and defines the inliers of S ;

Step 3: If the subset of S_i is greater than some threshold T , re-estimate the model using all of the points in S_i and terminate;

Step 4: If the size of S_i is less than T , select a new subset and repeat the above;

Step 5: After N trials, the largest consensus set S_i is selected, and the model is re-estimated using all of the points in the subset S_i .

return Curb model.

In Algorithm 2, the calculation of the distance is a key point, since there are two curb cases: straight line and curve; the distance calculation is different for these two cases. Fortunately, given the curb map obtained offline and the vehicle position, the shape of the curb can be known beforehand since the curb shape does not change much within a certain area. The following are the individual methods to calculate the distance.

A. Straight line curb case: In Figure 6, the positive direction of the X coordinate denotes the heading direction of the vehicle; the positive direction of the Y coordinate denotes the left side of the vehicle; the straight line denotes the curve, and its algebraic expression is:

$$ax + by + c = 0 \quad (24)$$

assume the coordinate of one curb point is (x_i, y_i) and the distance d_i is defined as the minimum distance from (x_i, y_i) to the straight line, which can be calculated as:

$$d_i = \frac{|ax_i + by_i + c|}{\sqrt{a^2 + b^2}} \quad (25)$$

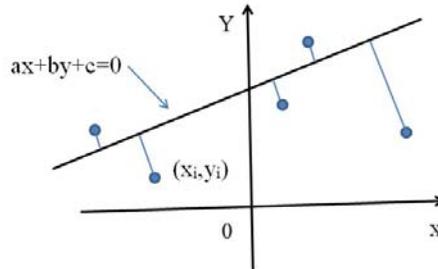


Figure 6. Straight line curb.

B. Curve line curb case: Similar to Figure 6, in Figure 7, the positive direction of the X coordinate denotes the heading direction of the vehicle; the positive direction of the Y coordinate denotes the left side of the vehicle. The curve line denotes the curb, and its algebraic expression is:

$$y = ax^2 + bx + c \quad (26)$$

In addition, each dashed line denotes the tangent line of the curve, which is perpendicular to the line joining the point of tangency and the curb point (x_i, y_i) . Assume that the coordinate of one curb point is (x_i, y_i) ; the distance d_i is also defined as the minimum distance from (x_i, y_i) to the curve, which can be calculated as follows:

Firstly, we need to find one point on the curve that is perpendicular to the line joining the point of tangency and the curb point (x_i, y_i) . Denote this point as $(x, ax^2 + bx + c)$; the slope of the tangent line across this point can be calculated as:

$$k_1 = 2a + b$$

and the slope joining this point and the curb point (x_i, y_i) is:

$$k_2 = \frac{y_i - (ax^2 + bx + c)}{x_i - x}$$

From $k_1 k_2 = -1$, we have:

$$A_i x^3 + B_i x^2 + C_i x + D_i = 0 \tag{27}$$

where:

$$\begin{aligned} A_i &= 2a^2, \\ B_i &= 3ab \\ C_i &= 2ac + b^2 + 1 - 2ay_i, \\ D_i &= bc - by_i - x_i \end{aligned}$$

The real root of (27) is:

$$x^* = -\frac{1}{3A_i} \left(B_i + C_i + \frac{\delta_0}{\delta_2} \right) \tag{28}$$

where:

$$\begin{aligned} \delta_0 &= B_i^2 - 3A_i C_i \\ \delta_1 &= 2B_i^3 - 9A_i B_i C_i + 27A_i^2 D_i \\ \delta_2 &= (0.5(\delta_1 + \sqrt{\delta_1^2 - 4\delta_0^3}))^{\frac{1}{3}} \end{aligned}$$

Then, we have:

$$y^* = ax^{*2} + bx^* + c \tag{29}$$

The minimum distance from curb point (x_i, y_i) to the curve is:

$$d_i = \sqrt{(x_i - x^*)^2 + (y_i - y^*)^2} \tag{30}$$

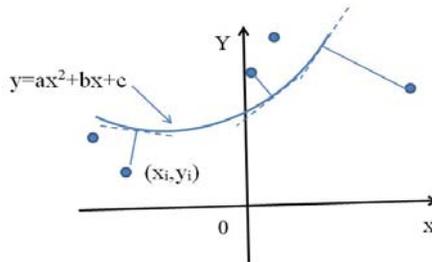


Figure 7. Curve line curb.

The following Algorithm 3 presents the pseudo-code of curb fitting.

Algorithm 3 Pseudo-code of curb fitting.**Require:**

M 3D points (only the X and Y coordinates are used);

Ensure:

Step 1: Initialize parameters of the algorithm. Let $N = 1000$; $T = 0.6 \times M$;

Step 2: Repeat for N iterations:

- a. Select two points randomly from the M points;
- b. Compute the parameters (a, b, c) that define the line passing through those two points;
- c. Count the number of inliers for the current line;
- d. If the number of inliers is greater than or equal to T , terminate;
- e. Keep the line having a maximum number of inliers;

Step 3: Draw the line having the maximum number of inliers.

return Line/curve parameters of curbs.

3.3. Lateral Correction Based on the Kalman Filter

3.3.1. Lateral Error Estimation

After fitting the curb into straight lines or curves (roundabout area) by using the candidate points selected from the point cloud of a 3D LiDAR, the next step is to calculate the lateral distance from the vehicle to the curb. Assume the curb is straight. The curve case is similar, as shown in Figure 8. The solid line denotes the RNDF curb. The dash-dotted line denotes the curb estimated. The origin denotes the position of the vehicle. In the lateral direction, i.e., the positive direction of the Y coordinate, there is a gap e , which is the lateral error caused by the localization error. Obviously, in order to correct this error, we need to adjust the vehicle position from the origin to the position of the rectangle on the dashed line, which means move the vehicle a distance e along the positive direction of the Y coordinate.

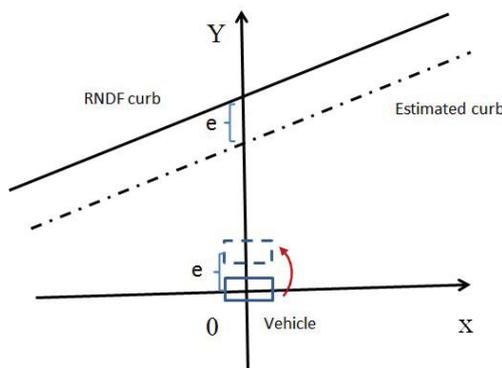


Figure 8. Lateral distances.

In order to filter the noise caused by sensor and measurement noise, etc., and smooth the lateral adjustment, a one-dimensional Kalman filter [29] is used to estimate the difference of lateral distances from the vehicle to the estimated curb and RNDF curb. Define the measurement vector y_t as the

difference e and the state vector x_t as the systematic part of this difference. Assume the change of x in time to be random. Set the system transition matrix as the identity matrix. We have that the system of equations takes the form:

$$x_t = x_{t-1} + w_t \quad (31)$$

the observation equation is given by:

$$y_t = x_t + v_t \quad (32)$$

where w_t, v_t are scalar variables of zero mean. The initial value x_0 is assumed to be zero. Then, our Kalman filter algorithm has the following form:

- State space: The space of real number \mathbf{R}
- State Vector: x_t
- System equation: $x_t = x_{t-1} + w_t$
- Observation: $y_t = x_t + v_t$
- Prediction equation: $x_{t/t-1} = x_t, P_{t/t-1} = P_{t-1} + W_t$
- Updating equations: $x_t = x_{t/t-1} + K_t(y_t - x_{t/t-1}),$

$$K_t = \frac{P_{t/t-1}}{P_{t/t-1} + V_t}$$

$$P_t = (1 - K_t)P_{t/t-1}$$

3.3.2. Lateral Adjustment

After we obtain the lateral difference e , which is the output of the one-dimensional Kalman filter as stated in the last subsection, next, we need to calculate the adjustment of the vehicle position in the global frame. Here we denote the adjustment as $(\Delta x, \Delta y)$.

Assume the global pose of the vehicle is (x_v, y_v, ϕ_v) , and the intersection point coordinate of the estimated curb line and Y coordinate in the local coordinate frame of the vehicle is (x_l, y_l) , while the global coordinate of this intersection point is denoted as (x_g, y_g) , then we have:

$$\Delta x = \frac{e}{d_{RNDF}}(x_g - x_v)$$

$$\Delta y = \frac{e}{d_{RNDF}}(y_g - y_v)$$

where e is the lateral distance error, d_{RNDF} is the lateral distance from vehicle to the RNDF curb line and:

$$x_g = x_v + x_l \cos(\phi_v) - y_l \sin(\phi_v)$$

$$y_g = y_v + x_l \sin(\phi_v) + y_l \cos(\phi_v)$$

Then, $(\Delta x, \Delta y)$ are added to the vehicle global position to correct the lateral error, i.e.,

$$x'_v = x_v + \Delta x$$

$$y'_v = y_v + \Delta y$$

where (x'_v, y'_v) is the vehicle coordinate after adjustment, which corresponds to the $[p^X, p^Y]$ of the vehicle position \mathbf{p} in (1).

4. Experimental Results

Result of GNSS/INS/DMI-Based Localization

In this subsection, we evaluate the performance of GNSS/IMU/DMI-based localization quantitatively using the publicly available dataset proposed in [30], where the ground truths of poses are provided to be 1.0 cm and 0.5° . Since the dataset does not include encoder data, artificial encoder data are generated from the RTK GNSS speed measurements, which are used as the ground truth. They are upsampled to get the output rate of 100 Hz. The proposed approach is tested on the experimental Campus-0L (5.563 min) in which the trajectory totally covers about 1143 m in distance with two sharp turns. In this paper, we compare the pose estimates from two UKF-based methods with the reference: one is the UKF method with the jump detection strategies, represented by 'UKF'; the other one is the UKF method without jump detection involved, represented by 'UKF w/o'. In what follows, we use 'REF' to represent the reference trajectory.

Figure 9 illustrates the estimated vehicle trajectories in the horizontal plane provided by the two methods against the reference. In the plot, the solid black line represents the reference trajectory; the dots indicate the GNSS measurements; the red dashed line shows the estimated trajectory by 'UKF'; and the blue dash-dotted line demonstrates the trajectories estimated by 'UKF w/o'. As we can see from the figure, the estimated trajectories from both methods follow the reference well, as there is no obvious jumps in this dataset.

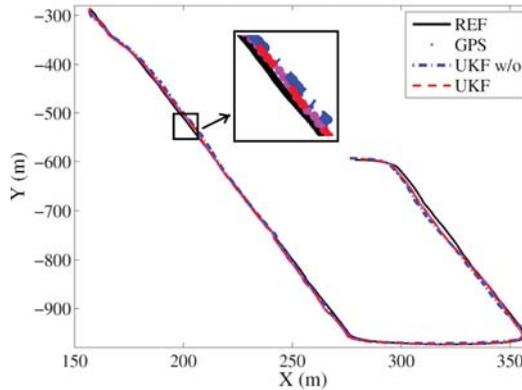


Figure 9. The estimated trajectories by the UKF-based methods with or without (w/o) the jump detection strategies against the reference.

During this experiment, no jump exists in the DGNSS measurements. To verify the efficiency of jump detection, some random noises were intentionally induced, which can be seen from the GNSS measurements represented by dots in Figure 10. Figure 10 illustrates the estimated trajectories in the horizontal plane against the reference with simulated GNSS jumps. The results are provided by the 'UKF w/o' and 'UKF' methods. As we can see from the figure, even with the GNSS jumps, the trajectory estimated by the 'UKF' method still follows the reference very well; however, the trajectory from the 'UKF w/o' method deviates from the reference when GNSS jumps are introduced.

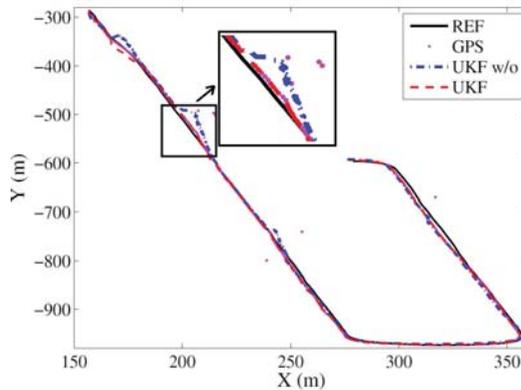


Figure 10. The estimated trajectories by the UKF-based methods with or without the jump detection strategies with random noises in the GNSS measurements against the reference.

From Figures 9 and 10, it can be seen that though the accuracy of position using UKF has been improved, lateral localization errors still exist, which may need to be corrected. To verify the lateral error correction method proposed in this paper, we apply it to the autonomous vehicle of the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore, which is equipped with GPS/IMU/DMI/LiDAR [31]; see Figure 11.



Figure 11. Autonomous vehicle equipped with GPS/IMU/DMI/LiDAR.

Point clouds of curb-like obstacles around the vehicle are detected by the 3D LiDAR mounted on the vehicle. After separating curbs from other obstacles and fitting them into straight lines and curve lines, lateral distances from vehicle to the curbs are calculated, then we apply the lateral correction method proposed in Section 3 to correct the lateral error of the GPS/IMU/DMI localization results. Figures 12 and 13 show the localization results of the autonomous vehicle without and with lateral error correction, respectively, where pink lines denote the curb detected using the method proposed in this paper, and the long yellow line segment in each figure is the curb line obtained from RNDF, while the short yellow line segment denotes the lateral distance between the vehicle to RNDF curbs. Figure 14 shows the lateral errors before and after lateral correction during a period of time. From Figures 12–14, it can be seen that lateral localization errors have been improved greatly.

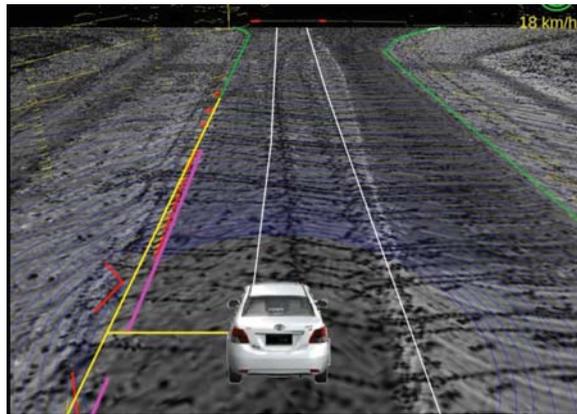


Figure 12. Localization result without lateral correction.

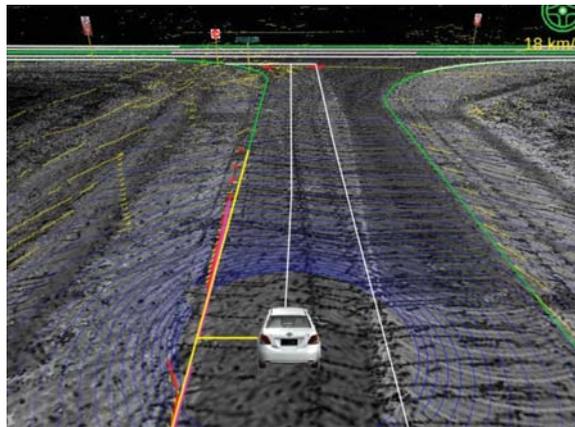


Figure 13. Localization result with lateral correction.

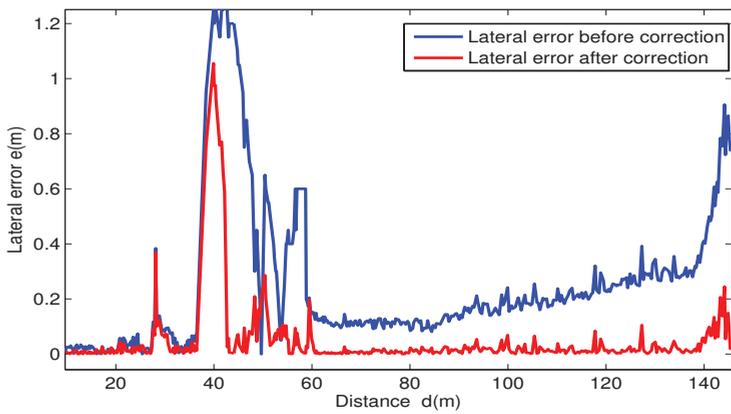


Figure 14. Lateral errors before and after lateral correction.

5. Conclusions

In this paper, to further improve the localization accuracy of an autonomous vehicle using the UKF-based GNSS/IMU/DMI fusion method, a multi-constraint fault-detection approach is proposed to handle the GNSS jumps. In addition, point cloud-based lateral correction is also proposed, where curbs are estimated in real time using a 3D LiDAR. A one-dimensional Kalman filter is adopted to estimate the lateral errors, which are used to improve lateral localization. Our method suppresses falsely-detected curb points by using the RANSAC algorithm. In future work, after the selection of point clouds for curbs, machine learning and deep learning algorithms may be used to further process the point clouds, such that curbs can be separated from other obstacles, such as vehicles and pedestrians.

Acknowledgments: This work was supported by the Funds of National Natural Science of China (Grant No. 61673098).

Author Contributions: Xiaoli Meng conceived and designed the experiments; Heng Wang wrote the paper and analyzed the data; Bingbing Liu performed the experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Montemerlo, M.; Becker, J.; Bhat, S.; Dahlkamp, H.; Dolgov, D.; Ettinger, S.; Haehnel, D.; Hilden, T.; Hoffmann, G.; Huhnke, B.; et al. Junior: The Stanford entry in the urban challenge. *J. Field Robot.* **2008**, *25*, 569–597.
2. Ronnback, S. Development of an INS/GPS Navigation Loop for an UAV. Master's Thesis, Lulea Tekniska Universitet, Lulea, Sweden, 2000.
3. Zhao, Y.M. *GPS/IMU Integrated System for Land Vehicle Navigation Based on MEMS*; Royal Institute of Technology: Stockholm, Sweden, 2011.
4. Pitt, M.K.; Shephard, N. Filtering via simulation: Auxiliary particle filters. *J. Am. Stat. Assoc.* **1999**, *94*, 590–599.
5. Arulampalam, S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filter for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188.
6. Nemra, A.; Aouf, N. Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering. *IEEE Sens. J.* **2010**, *10*, 789–797.
7. Vu, A.; Ramanandan, A.; Chen, A.; Farrell, J.A.; Barth, M. Real-time computer vision/DGPS-aided inertial navigation system for lane-level vehicle navigation. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 899–913.
8. Suhr, J.K.; Jang, J.; Min, D.; Jung, H.G. Sensor fusion-based low-cost vehicle localization system for complex urban environments. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1078–1086.
9. Oh, S.M. Multisensor fusion for autonomous UAV navigation based on the Unscented Kalman Filter with Sequential Measurement Updates. In Proceedings of the 2010 IEEE Conference Multisensor Fusion and Integration for Intelligent Systems (MFI), Salt Lake City, UT, USA, 5–7 September 2010; pp. 217–222.
10. Zhou, J.C.; Yang, Y.H.; Zhang, J.Y.; Edwan, E.; Loffeld, O. Tightly-coupled INS/GPS using Quaternion-based Unscented Kalman filter. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, USA, 8–11 August 2011.
11. Enkhtur, M.; Cho, S.Y.; Kim, K.H. Modified Unscented Kalman Filter for a multirate INS/GPS integrated navigation system. *ETRI J.* **2013**, *35*, 943–946.
12. Tao, Z.; Bonnifait, P.; Fremont, V.; Ibanez-Guzman, J. Mapping and localization using gps, lane markings and proprioceptive sensors. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 406–412.
13. Schindler, A. Vehicle self-localization with high-precision digital maps. In Proceedings of the IEEE Intelligent Vehicle Symposium, Gold Coast, Australia, 23–26 June 2013; pp. 141–146.
14. Gruyer, D.; Belaroussi, R.; Revilloud, M. Map-aided localization with lateral perception. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium (IV), Dearborn, MI, USA, 8–11 June 2014; pp. 674–680.

15. Pandey, G.; McBride, J.R.; Eustice, R.M. Ford campus vision and LiDAR data set. *Int. J. Rob. Res.* **2011**, *30*, 1543–1552.
16. Zhao, G.Q.; Yuan, J.S. Curb detection and tracking using 3D-LIDAR scanner. In Proceedings of the 19th IEEE International Conference on Image Processing (ICIP), Orlando, FL, USA, 30 September–3 October 2012.
17. Huang, A.S.; Teller, S. Lane boundary and curb estimation with lateral uncertainties. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 1729–1734.
18. Hervieu, A.; Soheilian, B. Road side detection and reconstruction using LIDAR sensor. In Proceedings of the 2013 IEEE on the Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 1247–1252.
19. Yeonsik, K.; Chiwon, R.; Seung-Beum, S.; Bongsob, S. A Lidar-based decision-making method for road boundary detection using multiple Kalman filters. *IEEE Trans. Ind. Electron.* **2012**, *59*, 4360–4368.
20. Kim, Z. Robust lane detection and tracking in challenging scenarios. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 16–26.
21. Kodagoda, K.; Wijesoma, W.; Balasuriya, A. CuTE: Curb tracking and estimation. *IEEE Trans. Control Syst. Technol.* **2006**, *14*, 951–957.
22. Siegemund, J.; Pfeiffer, D.; Franke, U.; Forstne, W. Curb reconstruction using conditional random fields. In Proceedings of the IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 21–24 June 2010.
23. Shin, Y.; Jung, C.; Chung, W. Drivable road region detection using a single laser range finder for outdoor patrol robots. In Proceedings of the IEEE Intelligent Vehicles Symposium(IV), San Diego, CA, USA, 21–24 June 2010.
24. Meng, X.; Zhang, Z.; Sun, S.; Wu, J.; Wong, W. Biomechanical model-based displacement estimation in micro-sensor motion capture. *Meas. Sci. Technol.* **2012**, *23*, 055101.
25. Kuipers, J. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*; Princeton University Press: Princeton, NJ, USA, 1999.
26. Choukroun, D.; Bar-Itzhack, I.; Oshman, Y. Novel quaternion Kalman filter. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 174–190.
27. Wan, E.A.; der Merwe, R.V. The Unscented Kalman Filter for Nonlinear Estimation. In Proceedings of the 2000 IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC'2000), Lake Louise, AB, Canada, 4 October 2000; pp. 153–158.
28. Sukkariéh, S.; Nebot, E.; Durrant-Whyte, H. A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Trans. Robot. Autom.* **1999**, *15*, 572–578.
29. Galanis, G.; Anadranistakis, M. A one-dimensional Kalman filter for the correction of near surface temperature forecasts. *Meteorol. Appl.* **2002**, *9*, 437–441.
30. Blanco, J.; Moreno, F.; Gonzalez, J. A Collection of Outdoor Robotic Datasets with centimeter-accuracy Ground Truth. *Auton. Robots* **2009**, *27*, 174–190.
31. Wang, H.; Wang, B.; Liu, B.B.; Meng, X.L.; Yang, G.H. Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle. *Robot. Auton. Syst.* **2017**, *88*, 71–78.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

An Improved Evidential-IOWA Sensor Data Fusion Approach in Fault Diagnosis

Yongchuan Tang *, Deyun Zhou *, Miaoyan Zhuang, Xueyi Fang and Chunhe Xie

School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China; zhuang-my@mail.nwpu.edu.cn (M.Z.); xyfang@nwpu.edu.cn (X.F.); xiechunhe@mail.nwpu.edu.cn (C.X.)

* Correspondence: tangyongchuan@mail.nwpu.edu.cn (Y.T.); dyzhou@nwpu.edu.cn (D.Z.)

Received: 30 July 2017; Accepted: 14 September 2017; Published: 18 September 2017

Abstract: As an important tool of information fusion, Dempster–Shafer evidence theory is widely applied in handling the uncertain information in fault diagnosis. However, an incorrect result may be obtained if the combined evidence is highly conflicting, which may lead to failure in locating the fault. To deal with the problem, an improved evidential-Induced Ordered Weighted Averaging (IOWA) sensor data fusion approach is proposed in the frame of Dempster–Shafer evidence theory. In the new method, the IOWA operator is used to determine the weight of different sensor data source, while determining the parameter of the IOWA, both the distance of evidence and the belief entropy are taken into consideration. First, based on the global distance of evidence and the global belief entropy, the α value of IOWA is obtained. Simultaneously, a weight vector is given based on the maximum entropy method model. Then, according to IOWA operator, the evidence are modified before applying the Dempster’s combination rule. The proposed method has a better performance in conflict management and fault diagnosis due to the fact that the information volume of each evidence is taken into consideration. A numerical example and a case study in fault diagnosis are presented to show the rationality and efficiency of the proposed method.

Keywords: Dempster–Shafer evidence theory; belief entropy; distance of evidence; IOWA operator; fault diagnosis; sensor data fusion

1. Introduction

The structure of the modern engineering system is more and more complex [1,2], and how to maintain the safety of these systems is a critical problem. Various types of faults may occur because of long-time continuous operation and the changing environmental factors, which may bring great threats to human life [3–6]. Therefore, fault diagnosis plays an important role in real applications in daily life [7–10]. In practical applications, a multi-sensor system is widely used in fault diagnosis to make a comprehensive judgment [11–13]. For example, fault detection and isolation have been successfully used on the well known Airbus aircraft [14,15], which plays a key role in ensuring the safety of the aircraft [16,17]. However, the information, which may be obtained from a multi-sensor system, is heterogeneous and imprecise [18]. Therefore, it is essential that the uncertain information is pre-processed before data fusion and decision-making [19,20].

Information fusion is a key technology of combining multi-source information [21,22]. To address the uncertain information, some mathematical tools focused on data fusion have been introduced, such as fuzzy sets theory [23,24], Dempster–Shafer evidence theory [25–27], comprehensive optimization algorithm [28,29] and so on [30–32]. As an important tool in information fusion, Dempster–Shafer evidence theory is widely applied in fault diagnosis [33], pattern recognition [34–37], multi-criteria decision-making [38–40], risk analysis [41–44], controller design [45,46] and so on [47–49]. However, an incorrect result may be obtained if the combined evidence is highly conflicting. To handle this problem, many methods have been presented [50–52].

In the frame of Dempster–Shafer evidence theory, while dealing with the conflicting data fusion, one kind of method is to modify the conventional combination rule. Yager modifies Dempster’s combination rule through redistributing the conflicting evidence [53]. However, this method may destroy the good properties of Dempster’s combination rule, such as the commutativity and associativity. In addition, it is unreasonable to blame the combination rule if the incorrect results are caused by sensor failure. Another typical method is to modify the evidence before applying Dempster’s combination rule. Murphy’s method averages the evidence, which does not consider the difference among the evidence [54]. The distance of evidence is used to obtain the weight in Deng et al.’s method [55], which does remedy the disadvantage of Murphy’s method to a certain extent.

In this paper, an improved evidential-Induced Ordered Weighted Averaging (IOWA) sensor data fusion method is proposed in dealing with multi-sensor data fusion in fault diagnosis. Firstly, according to the global distance of evidence d_g and the global belief entropy E_d^g , α value of the maximum entropy method (MEM) is established. Namely, the α value is jointly determined by d_g and E_d^g . Secondly, a weight vector $W = (w_1, w_2, \dots, w_n)^T$ is generated based on the MEM model. After that, the evidence are modified by the new IOWA-based weight factor. Finally, the obtained evidence is combined $(n - 1)$ times with Dempster’s combination rule. A numerical example and a case study on fault diagnosis verify the validity and reasonability of the proposed method.

This rest of this paper is organized as follows. The preliminaries are introduced in Section 2. In Section 3, a new evidential-IOWA sensor data fusion method is proposed. The application of the new method is presented in Section 4. Conclusions are given in Section 5.

2. Preliminaries

2.1. Dempster–Shafer Evidence Theory

Dempster–Shafer evidence theory was introduced by Dempster and then developed by Shafer, which is usually applied to manage the conflicting evidence [56,57].

Let Θ be the frame of discernment, and be defined as $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$. A basic probability assignment (BPA) $m : 2^\Theta \rightarrow [0, 1]$, is defined as follows [25,26]:

$$\begin{cases} m(\emptyset) = 0, \\ \sum_{A \subseteq \Theta} m(A) = 1. \end{cases} \tag{1}$$

when $m(A) > 0$, A is called a focal element.

Suppose m_1 and m_2 are two BPAs on the frame of discernment Θ , Dempster’s combination rule is defined as follows [25]:

$$m(A) = \begin{cases} \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - k}, & A \neq \emptyset, \\ 0, & A = \emptyset, \end{cases} \tag{2}$$

where $k = \sum_{B \cap C = \emptyset} m_1(B)m_2(C)$, is regarded as a measure of conflict between m_1 and m_2 . The larger the k , the larger the degree of conflict.

2.2. Jousselme Distance

Jousselme distance is presented to measure of the difference—or the lack of similarity—between any two BPAs, which is introduced as follows.

Let m_1 and m_2 be two BPAs on the frame of discernment Θ , then the distance between m_1 and m_2 is [58]:

$$d(m_1, m_2) = \sqrt{\frac{1}{2}(\vec{m}_1 - \vec{m}_2)^T \underline{D}(\vec{m}_1 - \vec{m}_2)}, \tag{3}$$

where \underline{D} is an $2^{|\Theta|} \times 2^{|\Theta|}$ matrix whose elements are

$$D(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad A, B \in 2^\Theta. \tag{4}$$

2.3. Belief Entropy

Deng entropy is the generalization of Shannon entropy [59], which is defined as follows [60]:

$$E_d = - \sum_i m(B_i) \log_2 \frac{m(B_i)}{2^{|B_i|} - 1}, \tag{5}$$

where B_i is a proposition in the BPAs, and $|B_i|$ is the cardinality of B_i .

The entropy can definitely degenerate to the Shannon entropy especially when the belief is only assigned to single element. Namely,

$$E_d = - \sum_i m(C_i) \log_2 \frac{m(C_i)}{2^{|C_i|} - 1} = - \sum_i m(C_i) \log_2 m(C_i), \tag{6}$$

and, for $m_1(A) = \frac{2^{|A|} - 1}{\sum_{B \subseteq X} 2^{|B|} - 1}$, $A, B \subseteq X$, m_1 is the mass function having the maximum Deng entropy for the frame of discernment $X = \{a, b, c\}$, and its uncertainty can also be calculated by $\sum_{B \subseteq X} \log_2(2^{|B|} - 1)$.

2.4. IOWA Operator

The Induced Ordered Weighted Averaging (IOWA) operator [61], which is introduced by Yager and Filev, is a more general type of the Ordered Weighted Averaging (OWA) operator. An important feature of this operator is that the ordering of the arguments is induced by another variable called the order inducing variable.

Assume there are n two-tuple OWA pair $\langle u_i, a_i \rangle$, $i = 1, \dots, n$ that has an associated weight vector $W = (w_1, w_2, \dots, w_n)^T$ of dimension n having the following properties:

$$\begin{aligned} 0 \leq w_j \leq 1, \\ \sum_{j=1}^n w_j = 1. \end{aligned} \tag{7}$$

Then, the IOWA operator is defined as follows [61]:

$$F_w(\langle u_1, a_1 \rangle, \dots, \langle u_i, a_i \rangle) = \sum_{j=1}^n w_j b_j, \tag{8}$$

where b_j is the a_i of the OWA pair having the j th largest u_i . u_i is referred as the order inducing variable and a_i is referred as the argument variable.

orness, which is associated with the weight vector $W = (w_1, w_2, \dots, w_n)^T$, is defined as follows:

$$\alpha = orness(W) = \frac{1}{n-1} \sum_{j=1}^n w_j (n-j), \tag{9}$$

where $0 \leq orness \leq 1$.

2.5. Maximum Entropy Method

To apply the IOWA operator in fault diagnosis, a very crucial issue is to determine its weight. The weight problem is denoted as a constrained nonlinear optimization model in the MEM model, which is presented by O’Hagan. The weight is gained by the following optimization model [62]:

$$\begin{aligned}
 & \text{Maximize : } Disp(W) = - \sum_{j=1}^n w_j \ln(w_j), \\
 & \text{S.t } orness(W) = \alpha = \frac{1}{n-1} \sum_{j=1}^n w_j(n-j), \\
 & \sum_{j=1}^n w_j = 1, \\
 & 0 \leq w_j \leq 1 \text{ for } j = 1, \dots, n.
 \end{aligned}
 \tag{10}$$

Suppose $n = 5$ and the weights satisfy different degrees of *orness* : $\alpha = 0, 0.1, \dots, 1$, then the weight vector is determined by MEM model, which is shown in Figure 1.

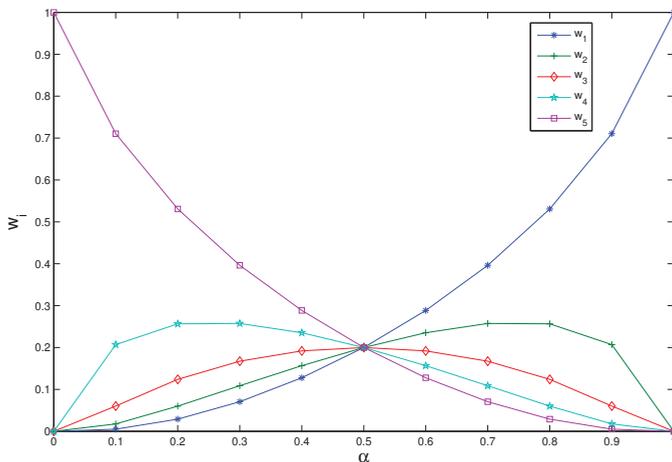


Figure 1. Variation of the weight with orness degree.

From Figure 1, we can conclude that: the value of the weight vector is closer to the average value $W = (1/n, 1/n, \dots, 1/n)^T$; the value of α is closer to $\alpha = 0.5$; the value of the weight vector is closer to $W = (1, 0, \dots, 0)^T$; the value of α is closer to $\alpha = 1$. Namely, the smaller the credibility gap among BPAs, the more average for weight distribution.

3. The Evidential IOWA-Based Fault Diagnosis Method

As shown in Figure 2, in the fault diagnosis technique, typically, the first step should be information collecting from actuators. Secondly, all hypotheses are modelled (by BPAs in the frame of Dempster–Shafer evidence theory). Thirdly, the evidence is modified according to the IOWA operator. Finally, data fusion is applied for fault diagnosis and decision-making. Here, how to get an appropriate weight to modify the evidence is very important for locating the possible fault accurately. In the proposed method, the MEM model based on the distance of evidence and the belief entropy are used to generate the appropriate weight of evidence.

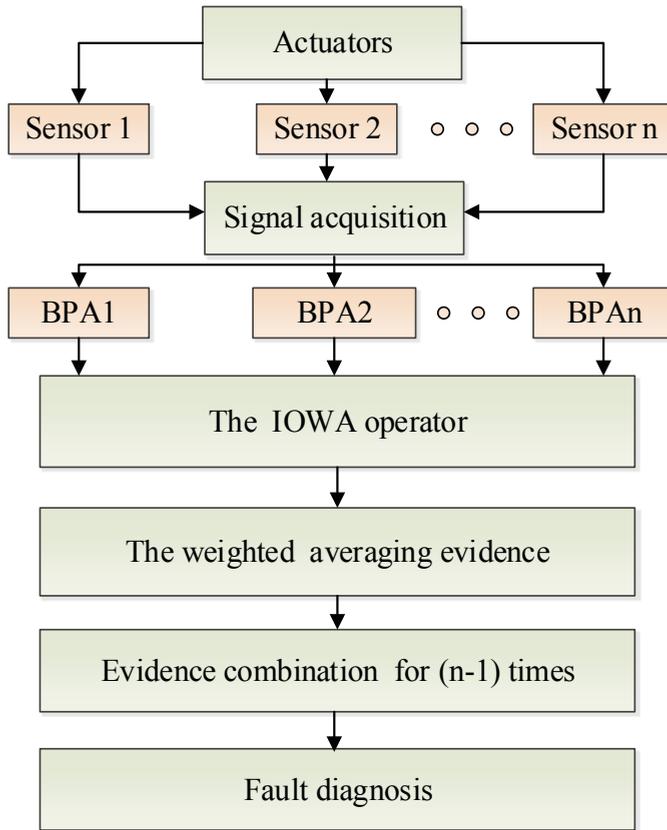


Figure 2. Overall structure of fault diagnosis based on sensor data fusion.

3.1. The Evidential-IOWA Parameter

Recently, the IOWA operator has aroused the attention of scholars and is widely used in real applications [63–65]. However, there are some problems while using the IOWA operator. For example, the α value of a constraint condition usually depends on the experience of the experts, which does not lead to an objective result. In this paper, based on the the distance of evidence and the belief entropy, the α value is induced as an objective weight.

3.1.1. Definition of α in IOWA

The distance of evidence and the belief entropy are jointly considered to determine the α value. The value of α is defined as follows:

$$\alpha = \frac{1}{2}(\alpha_1 + \alpha_2) = \frac{1}{2}(e^{d_g \cdot \ln 0.5} + 0.5E_d^g), \tag{11}$$

where d_g is the global distance of evidence, E_d^g is the global belief entropy, and $0 \leq d_g \leq 1$, $0 \leq E_d^g \leq 1$, $0.5 \leq \alpha \leq 1$. α_1 is a data-driven value based on the distance of evidence, and α_2 is another data-driven value based on belief entropy.

3.1.2. Definition of α_1 Based on the Distance of Evidence

Assume that there are many pieces of evidence for fault diagnosis. The Jousselme distances d_{ij} , $i, j = 1, 2, \dots, n$ between two evidence m_i and m_j can be calculated according to Equation (3), and the distance matrix (DM) is defined as follows:

$$DM = [d_{ij}] = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{bmatrix}. \tag{12}$$

The average distance of evidence of m_i , $i = 1, 2, \dots, n$, with respect to the other evidence, denoted as \bar{d}_i , is defined as follows:

$$\bar{d}_i = \frac{\sum_{i=1, i \neq j}^n d_{ij}}{n-1}, \quad i = 1, 2, \dots, n, \tag{13}$$

then, the global distance of evidence among all the evidence d_g is defined as follows:

$$d_g = \frac{\sum_{i=1}^n \bar{d}_i}{n}, \quad i = 1, 2, \dots, n. \tag{14}$$

If the global distance of evidence d_g has a big value, the smaller the global similarity degree among the diagnosed results, the smaller the credibility degree of each sensor. In other words, the smaller the weight gap among the BPAs, the more average the weight distribution is, which means that the value of α is closer to $\alpha = 0.5$. If $d_g = 1$, which means that the diagnosed fault type of multi-sensor is entirely different; in this case, the credibility degree of each evidence is the same with each other. Thus, the evidence should be assigned the same weight, namely, the weight vector is $W = (1/n, 1/n, \dots, 1/n)^T$ and $\alpha = 0.5$.

Conversely, the smaller the value of d_g , the greater the global similarity degree of the diagnosed results, so the BPAs can be represented approximately by less or even one BPA with a high credibility degree. That is to say, the BPA with high credibility degree is given a greater weight and the BPA with a low credibility degree is given a small weight. Thus, the smaller the value of d_g , the more inequality of the weight distribution, which means the value of α is closer to $\alpha = 1$. If $d_g = 0$, which means that the diagnosed results are similar, so the BPA can be represented by any BPAs. Considering the consistency of the algorithm, the initial weight is assigned as $W = (1, 0, \dots, 0)^T$ and $\alpha = 1$.

Based on the above analysis, a relational formula of the degree of *orness* α_1 is defined as follows:

$$\alpha_1 = e^{d_g \cdot \ln 0.5}, \tag{15}$$

where d_g is the global distance of evidence, and $0 \leq d_g \leq 1$, $0.5 \leq \alpha_1 \leq 1$.

3.1.3. Definition of α_2 Based on the Belief Entropy

Deng entropy is an efficient tool to measure uncertainty, not only under the situation where the uncertainty is represented by a probability distribution, but also under the situation where the uncertainty is represented by the BPAs. Thus, this entropy is used to determine the α value.

The global belief entropy E_d^g is defined as follows:

$$E_d^g = \frac{\sum_{i=1}^n E_{d_i}}{n \cdot (E_d)_{\max}}, \tag{16}$$

where E_{d_i} is the belief entropy of the evidence m_i . $(E_d)_{\max}$ is the maximum belief entropy on the frame of discernment X , which is defined as:

$$(E_d)_{\max} = \log \sum_{B \subseteq X} (2^{|B|} - 1). \tag{17}$$

The greater the global belief entropy E_d^g , the greater the global uncertainty of the diagnosed faults. Therefore, the weight distribution should be more average, and the α value is more close to 0.5. If $E_d^g = 1$, it shows that the diagnosed faults is entirely uncertainty, so they should be assigned to the same weight, that is, $\alpha = 0.5$.

The smaller the global belief entropy E_d^g , the smaller the global uncertainty of the diagnosed faults. Then, the BPA can be represented approximately by a few or even one BPA of relatively small uncertainty. Therefore, the smaller the E_d^g , the more inequality the weight distribution, the closer $\alpha = 1$. If $E_d^g = 0$, the BPA can be represented by any BPAs, that is to say, the weight vector is $W = (1, 0, \dots, 0)^T$ and $\alpha = 1$.

Based on the above analysis, a relational formula of the degree of *orness* α_2 is defined as follows:

$$\alpha_2 = 0.5^{E_d^g}, \tag{18}$$

where E_d^g is the global belief entropy, and $0 \leq E_d^g \leq 1, 0.5 \leq \alpha_2 \leq 1$.

3.1.4. The Weight Vector of IOWA

After obtaining the parameters α , the weight vector $W = (w_1, w_2, \dots, w_n)^T$ can be obtained according to the MEM model. Assume that there are n BPAs $m_i, i = 1, 2, \dots, n$, the weight vector $W = (w_1, w_2, \dots, w_n)^T$ can be calculated according to the following steps:

- Step 1** According to Equations (14) and (15), the global distance of evidence d_g and the α_1 value can be calculated, respectively.
- Step 2** The global belief entropy E_d^g and the α_2 value are obtained by Equations (16) and (18), respectively.
- Step 3** The α value and the weight vector W are calculated based on Equations (11) and (10), respectively.

3.2. Multi-Evidential Fusion Model

After getting an appropriate weight vector, the evidence can be modified before using Dempster’s combination rule. The evidence are reordered according to the IOWA operator. Assume there are n BPAs, denoted as $m_i, i = 1, 2, \dots, n$, the steps of ordering and evidence fusion are defined as follows:

- Step 1** Construct the inducing variable S_i :

$$S_i = 1 - \bar{d}_i, i = 1, 2, \dots, n, \tag{19}$$

where \bar{d}_i is the average distance of evidence obtained by Equation (13).

- Step 2** Obtain the OWA pairs $\langle S_i, M_i \rangle, i = 1, 2, \dots, n$, where M_i is the argument variable, namely, it is the BPAs of the evidence m_i .
- Step 3** According to Equation (8), the weighted average evidence can be calculated.
- Step 4** Combine the new evidence with Dempster’s combination rule by $(n - 1)$ times.

With the fusion results, decision-making can be made based on the maximum principle of BPAs. An illustrative explanation of the new method is presented in Figure 3. Firstly, the degree of *orness* α should be computed based on distance of evidence and belief entropy. Secondly, the weight vector $W = (w_1, w_2, \dots, w_n)^T$ can be obtained based on the MEM model. Thirdly, a corresponding inducing variable can be constructed. Fourthly, evidence modification and fusion can be achieved. Finally, decision-making in fault diagnosis is based on the fused results.

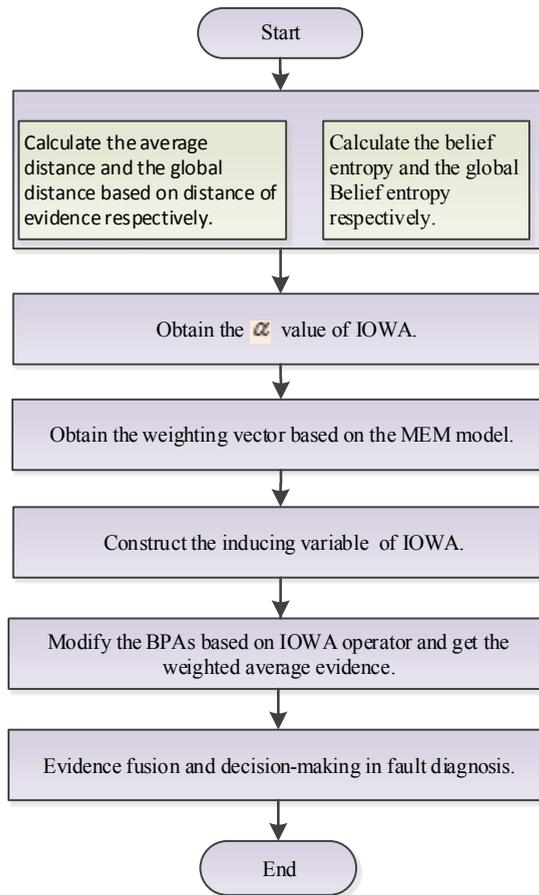


Figure 3. The evidential Induced Ordered Weighted Averaging (IOWA)-based fault diagnosis method.

4. Application

4.1. Experiment with Artificial Data

This numerical example is used to illustrate how to apply the proposed method in fault diagnosis. Assume that, in the case of motor rotor fault diagnosis, vibration signal is collected by five sensors. There are three faults, denoted as A , B and C , in motor rotor, which represents the unbalance, misalignment and pedestal looseness fault types, respectively. The BPAs based on these sensors are assumed to be independent and there are abnormal sensor reports, as is shown in Table 1. Intuitively, m_2 comes from abnormal sensor report. Since evidence modelling is another open issue in Dempster–Shafer evidence theory, we do not discuss how to model data with BPAs in this paper. For more detail on how to generate BPAs, please refer to some related work such as [45,46,49].

Table 1. The basic probability assignment (BPA) as an example.

m_i	{A}	{B}	{C}	{AC}
m_1	0.41	0.29	0.30	0.00
m_2	0.00	0.90	0.10	0.00
m_3	0.58	0.07	0.00	0.35
m_4	0.55	0.10	0.00	0.35
m_5	0.6	0.10	0.00	0.30

According to the new method shown in Figure 3, firstly, with Equations (13) and (14), the average distance of evidence $\bar{d}_i, i = 1, 2, \dots, 5$ and the global distance of evidence d_g can be calculated, respectively, and the results are: $\bar{d}_1 = 0.3456, \bar{d}_2 = 0.6647, \bar{d}_3 = 0.2661, \bar{d}_4 = 0.2564, \bar{d}_5 = 0.2641$ and $d_g = 0.3594$. With Equations (5) and (16), the belief entropy $E_{di}, i = 1, 2, \dots, 5$ and the global belief entropy E_d^g can be calculated, respectively, and the results are: $E_{d1} = 1.5664, E_{d2} = 0.4690, E_{d3} = 1.8092, E_{d4} = 1.8914, E_{d5} = 1.7710$ and $E_d^g = 0.3534$.

Secondly, the degree of orness α can be calculated by Equation (11):

$$a = \frac{1}{2}(e^{0.3594 \cdot \ln 0.5} + 0.5^{0.3534}) = 0.7811. \tag{20}$$

The weight vector $W = (w_1, w_2, w_3, w_4, w_5)^T$ is calculated according to Equation (10), and the result is:

$$W = (0.5026, 0.2592, 0.1337, 0.0689, 0.0356)^T. \tag{21}$$

In addition, the inducing variable $S_i, i = 1, 2, \dots, 5$ are calculated according to Equation (19):

$$\begin{aligned} S_1 &= 1 - 0.3456 = 0.6544, \\ S_2 &= 1 - 0.6647 = 0.3353, \\ S_3 &= 1 - 0.2661 = 0.7339, \\ S_4 &= 1 - 0.2564 = 0.7436, \\ S_5 &= 1 - 0.2641 = 0.7359. \end{aligned} \tag{22}$$

Thirdly, according to the ordering variable \bar{d}_i , the OWA pair $\langle S_i, M_i \rangle, i = 1, 2, \dots, 5$ are ordered as follows:

$$\begin{aligned} &\langle 0.7436, M_4 \rangle, \\ &\langle 0.7359, M_5 \rangle, \\ &\langle 0.7339, M_3 \rangle, \\ &\langle 0.6544, M_1 \rangle, \\ &\langle 0.3353, M_2 \rangle. \end{aligned} \tag{23}$$

Then, the BPAs on each fault are modified according to Equation(8), and the weighted average evidence is:

$$m(A) = 0.5411, \quad m(B) = 0.1325, \quad m(C) = 0.0217, \quad m(AC) = 0.3047. \tag{24}$$

Finally, combining the weighted average evidence with Dempster’s combination rule by four times, the final results are shown as follows:

$$m(A) = 0.9914, \quad m(B) = 0.0001, \quad m(C) = 0.0025, \quad m(AC) = 0.0061. \tag{25}$$

In Table 2, we compare the results among several existing methods. It also shows the process of locating the fault type. With the new method, the belief in the fault diagnosis results that *A* is the fault type is 99.14%, which is not lower than the other methods.

Table 2. Comparison of several existing methods.

BPA	Methods	$m(A)$	$m(B)$	$m(C)$	$m(AC)$	Faults
m_1, m_2	Dempster’s method [25]	0	0.8969	0.1031	0	<i>B</i>
	Murphy’s method [54]	0.0964	0.8119	0.0917	0	<i>B</i>
	Deng et al.’s method [55]	0.0964	0.8119	0.0917	0	<i>B</i>
	The proposed method	0.0964	0.8119	0.0917	0	<i>B</i>
m_1, m_2, m_3	Dempster’s method [25]	0	0.6350	0.3650	0	<i>B</i>
	Murphy’s method [54]	0.4939	0.4180	0.0792	0.0090	<i>A</i>
	Deng et al.’s method [55]	0.4974	0.4054	0.0888	0.0084	<i>A</i>
	The proposed method	0.6960	0.1744	0.1253	0.0056	<i>A</i>
m_1, m_2, m_3, m_4	Dempster’s method [25]	0	0.3321	0.6679	0	<i>C</i>
	Murphy’s method [54]	0.8362	0.1147	0.0410	0.0081	<i>A</i>
	Deng et al.’s method [55]	0.9089	0.0444	0.0379	0.0089	<i>A</i>
	The proposed method	0.9683	0.0020	0.0133	0.0163	<i>A</i>
m_1, m_2, m_3, m_4, m_5	Dempster’s method [25]	0	0.1422	0.8578	0	<i>C</i>
	Murphy’s method [54]	0.9620	0.0210	0.0138	0.0032	<i>A</i>
	Deng et al.’s method [55]	0.9820	0.0039	0.0107	0.0034	<i>A</i>
	The proposed method	0.9914	0.0001	0.0025	0.0061	<i>A</i>

In addition, if Dempster’s combination rule is used directly, due to the conflicting evidence m_2 , incorrect results are obtained. The same diagnosis results can be obtained according to the Murphy’s method [54], Deng et al.’s method [55] and the proposed method. However, Murphy’s method is only a simple arithmetic mean which does not consider the difference among the evidence, while Deng et al.’s method ignores the influence of evidence itself in generating the weight factor. The proposed method takes into consideration more available information before making data fusion and fault diagnosis, e.g., the distance of evidence and the belief entropy.

4.2. A Case Study

In order to verify the effectiveness and success of the proposed evidential-IOWA sensor data fusion approach, the new method is applied to a case study adopted from [66].

Recall the fault diagnosis problem in [66]. Three potential fault types are denoted as F_1, F_2 and F_3 ; thus, the fault hypothesis set is $\Theta = \{F_1, F_2, F_3\}$. Three sensors report the diagnosis results independently, the diagnosis results are modelled as three bodies of evidence, denoted as E_1, E_2 and E_3 , and the BPAs of the diagnosis results are shown in Table 3. Intuitively, F_1 is the fault type because both E_1 and E_3 have a belief of more than 60% on the fault type F_1 , while the E_2 may come from an abnormal sensor in comparison with the other two bodies of evidence. This is a challenge for data fusion, especially for some conventional combination rules, such as Dempster’s rule of combination. The proposed method is applied to solve this problem.

Table 3. BPAs for fault diagnosis of the case study [66].

Sensor Report	$\{F_1\}$	$\{F_2\}$	$\{F_2, F_3\}$	Θ
$E_1 : m_1(\cdot)$	0.60	0.10	0.10	0.20
$E_2 : m_2(\cdot)$	0.05	0.80	0.05	0.10
$E_3 : m_3(\cdot)$	0.70	0.10	0.10	0.10

According to the proposed method shown in Figure 3, the first step is to calculate the average distance and global distance of the evidence E_1, E_2 and E_3 . Based on Equations (13) and (14), the calculation results of the average distance of each piece of evidence, denoted as $\bar{d}_i(E_i), i = 1, 2, 3$, and the global distance, denoted as $d_g(E_i), i = 1, 2, 3$, are shown in Table 4.

Table 4. The average distance and global distance of $E_i (i = 1, 2, 3)$.

Evidence Distance-Based Parameter	$\bar{d}_1(E_1)$	$\bar{d}_2(E_2)$	$\bar{d}_3(E_3)$	$d_g(E_i)$
Calculation Result	0.1916	0.3477	0.2033	0.3712

Then, based on Equations (5) and (16), the corresponding belief entropy, denoted as $E_{di}(E_i), i = 1, 2, 3$, and the global belief entropy, denoted as $E_d^g(E_i), i = 1, 2, 3$, are calculated in Table 5.

Table 5. The belief entropy and global belief entropy of $E_i (i = 1, 2, 3)$.

Belief Entropy-Based Parameter	$E_{d1}(E_1)$	$E_{d2}(E_2)$	$E_{d3}(E_3)$	$E_d^g(E_i)$
Calculation Result	2.2909	1.3819	1.7960	0.5884

With Equation (11), the degree of *orness* α of the case study, denoted as $\alpha(E_i)$, is calculated as follows:

$$\alpha(E_i) = \frac{1}{2}(e^{0.3712 \cdot \ln 0.5} + 0.5^{0.5884}) = 0.7189. \tag{26}$$

According to the Maximum Entropy Method defined in Equation (10), the weight vector of the evidence, denoted as $W(E_i) = (w_1, w_2, w_3)^T$, can be calculated, and the result is

$$W(E_i) = (0.5771, 0.2836, 0.1393)^T. \tag{27}$$

The inducing variable, denoted as $S_i(E_i) (i = 1, 2, 3)$, can be calculated based on Equation (19) and the parameters in Table 4, and the results are shown as follows:

$$\begin{aligned} S_1(E_1) &= 1 - 0.1916 = 0.8084, \\ S_2(E_2) &= 1 - 0.3477 = 0.6523, \\ S_3(E_3) &= 1 - 0.2033 = 0.7967. \end{aligned} \tag{28}$$

Combining the inducing variables with the parameters in Table 5, the OWA pairs $\langle S_i(E_i), E_i \rangle, i = 1, 2, 3$, are ordered as follows:

$$\begin{aligned} &\langle 0.8084, E_1 \rangle, \\ &\langle 0.7967, E_3 \rangle, \\ &\langle 0.6523, E_2 \rangle. \end{aligned} \tag{29}$$

Now, the BPAs in Table 3 can be modified according to Equation (8), and the weighted average evidence are as follows:

$$m(F_1) = 0.5517, \quad m(F_2) = 0.1975, \quad m(F_2, F_3) = 0.0930, \quad m(\Theta) = 0.1577. \tag{30}$$

Finally, combining the weighted average evidence with Dempster’s combination rule by four times, the fusion results are as follows:

$$m(F_1) = 0.9123, \quad m(F_2) = 0.0810, \quad m(F_2, F_3) = 0.0027, \quad m(\Theta) = 0.0039. \tag{31}$$

The fused results with the proposed method are compared with the method in [66] where this case study comes from, and the comparison result is shown in Table 6.

Table 6. Fusion results with different methods.

Fault Types	$\{F_1\}$	$\{F_2\}$	$\{F_2, F_3\}$	Θ
Only Dempster's Rule of Combination	0.4519	0.5048	0.0336	0.0096
Fan et al's Method [66]	0.8119	0.1096	0.0526	0.0259
The Proposed Method	0.9123	0.0810	0.0027	0.0039

It can be concluded from Table 6 that the proposed method has the most distinguishable fusion results on sensor reports, which means a clear indicator on the most possible fault type. The highest belief degree on fault type F_1 is 91.23%, which is higher than the method with Fan et al's method with more than 10%. This is helpful for decision-making in real applications. While the fusion results of fault type F_1 and F_2 with the conventional Dempster's rule of combination are close to each other, it is hard to judge which fault has occurred. The case study verifies the effectiveness of the proposed method. In addition, the case study indicates a better performance of the proposed method in comparison with some of the existing methods.

4.3. Discussion

The effectiveness of the proposed method is verified according to the applications based on both artificial data and the experiment adopted from the literature.

A few reasons contribute to the success of the new method. Firstly, not only the distance of evidence, but also the belief entropy and the IOWA operator are taken into consideration, which means more available information are used while doing information processing. Thus, information loss is decreased. Secondly, the way of getting the degree of *orness* α of IOWA (based on belief entropy and evidence distance) is data-driven, which is more reliable compared with some subjective methods. Finally, the final fused rule is based on Dempster's rule of combination. The merits of Dempster's rule of combination, such as satisfying the commutativity and associativity, contribute to the effectiveness of the proposed method.

In the fault diagnosis (FD) research area, an FD technique is good if a new method can guarantee that there is no false alarm, no missed detection and a full detection for all considered faulty scenarios [67,68]. The ongoing work of the proposed method should try to focus on this case. In future work, the following situations should be well addressed:

- FD without fault to be sure that the proposed solution doesn't give false alarm,
- FD with a misalignment fault to highlight that we detect this fault well,
- FD with pedestal fault.

5. Conclusions

In this paper, in the frame of Dempster–Shafer evidence theory, an improved evidential-IOWA sensor data fusion approach is proposed in dealing with a multi-source data-based fault diagnosis problem. Before applying sensor data fusion for final decision-making, the sensor data comes from different independent sources modelled, as BPA is pre-processed to avoid unreasonable fusion results that may be caused by conflicting evidence. In the new method, the IOWA operator is used to determine the weight of different sensor data sources, and the parameter of the IOWA is based on the distance of evidence and the belief entropy. The proposed method has a better performance in conflict management and fault diagnosis due to the fact that the information volume of each piece of evidence is taken into consideration. The proposed method outperforms the other methods according to the applications.

The ongoing work of the proposed method will be focused on some basic rules of fault diagnosis in industrial environmental scenarios, e.g., no missed detection and a full detection for all considered faulty scenarios should be strictly obeyed while applying the fault diagnosis technique.

Acknowledgments: The authors greatly appreciate the editor's encouragement and the two anonymous reviewers' valuable comments and suggestions to improve this paper. The work is partially supported by the National Natural Science Foundation of China (Grant No. 61671384), the Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2016JM6018) and the Innovation Foundation for the Doctor Dissertation of Northwestern Polytechnical University (Grant No. CX201705).

Author Contributions: Y.T. and D.Z. designed and performed the research. Y.T. and M.Z. wrote the manuscript. Y.T., M.Z. and C.X. performed the computation. X.F. analyzed the data. All authors discussed the results and commented on the manuscript.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Patecornell, M. Organizational aspects of engineering system safety: The case of offshore platforms. *Science* **1990**, *250*, 1210–1217.
2. Oosterom, M.; Babuska, R.; Verbruggen, H.B. Soft computing applications in aircraft sensor management and flight control law reconfiguration. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2002**, *32*, 125–139.
3. Picazo-Rodenas, M.J.; Royo, R.; Antonino-Daviu, J.; Roger-Folch, J. Use of the infrared data for heating curve computation in induction motors: Application to fault diagnosis. *Eng. Fail. Anal.* **2013**, *35*, 178–192.
4. Zhang, B.; Tan, A.C.C.; Lin, J.H. Gearbox fault diagnosis of high-speed railway train. *Eng. Fail. Anal.* **2016**, *66*, 407–420.
5. Du, Y.; Lu, X.; Su, X.; Hu, Y.; Deng, Y. New failure mode and effects analysis: An evidential downscaling method. *Qual. Reliab. Eng. Int.* **2016**, *32*, 737–746.
6. Chen, C.; Brown, D.; Sconyers, C.; Zhang, B.; Vachtsevanos, G.; Orchard, M.E. An integrated architecture for fault diagnosis and failure prognosis of complex engineering systems. *Expert Syst. Appl.* **2012**, *39*, 9031–9040.
7. Ferreira de Loza, A.; Cieslak, J.; Henry, D.; Zolghadri, A.; Fridman, L.M. Output tracking of systems subjected to perturbations and a class of actuator faults based on HOSM observation and identification. *Automatica* **2015**, *59*, 200–205.
8. Jiang, W.; Xie, C.; Zhuang, M.; Shou, Y.; Tang, Y. Sensor Data Fusion with Z-Numbers and Its Application in Fault Diagnosis. *Sensors* **2016**, *16*, 1509.
9. Su, X.; Deng, Y.; Mahadevan, S.; Bao, Q. An improved method for risk evaluation in failure modes and effects analysis of aircraft engine rotor blades. *Eng. Fail. Anal.* **2012**, *26*, 164–174.
10. Oukhellou, L.; Debiolles, A.; Denoeux, T.; Aknin, P. Fault diagnosis in railway track circuits using Dempster-Shafer classifier fusion. *Eng. Appl. Artif. Intell.* **2010**, *23*, 117–128.
11. Sun, S.; Lin, H.; Ma, J.; Li, X. Multi-sensor distributed fusion estimation with applications in networked systems: A review paper. *Inf. Fusion* **2017**, *38*, 122–134.
12. Pang, C.; Sun, S. Fusion Predictors for Multisensor Stochastic Uncertain Systems With Missing Measurements and Unknown Measurement Disturbances. *IEEE Sens. J.* **2015**, *15*, 4346–4354.
13. De Loza, A.F.; Cieslak, J.; Henry, D.; Davila, J.; Zolghadri, A. Sensor fault diagnosis using a non-homogeneous high-order sliding mode observer with application to a transport aircraft. *IET Control Theory Appl.* **2015**, *9*, 598–607.
14. Berdjag, D.; Cieslak, J.; Zolghadri, A. Fault diagnosis and monitoring of oscillatory failure case in aircraft inertial system. *Control Eng. Pract.* **2012**, *20*, 1410–1425.
15. Berdjag, D.; Zolghadri, A.; Cieslak, J.; Goupil, P. Fault detection and isolation for redundant aircraft sensors. In proceedings of 2010 Conference on Control and Fault-Tolerant Systems (SysTol), Nice, France, 6–8 October 2010; pp. 137–142.
16. Cieslak, J.; Henry, D.; Zolghadri, A.; Goupil, P. Development of an active fault-tolerant flight control strategy. *J. Guid. Control Dyn.* **2008**, *31*, 135–147.

17. Efimov, D.; Cieslak, J.; Zolghadri, A.; Henry, D. Actuator fault detection in aircraft systems: Oscillatory failure case study. *Annu. Rev. Control* **2013**, *37*, 180–190.
18. Li, C.; Mahadevan, S.; Ling, Y.; Choe, S.; Wang, L. Dynamic Bayesian Network for Aircraft Wing Health Monitoring Digital Twin. *AIAA J.* **2017**, *55*, 930–941.
19. He, Y.; Hu, L.F.; Guan, X.; Deng, Y.; Han, D. New method for measuring the degree of conflict among general basic probability assignments. *Sci. Chin.-Inf. Sci.* **2012**, *55*, 312–321.
20. Tang, Y.; Zhou, D.; Xu, S.; He, Z. A Weighted Belief Entropy-Based Uncertainty Measure for Multi-Sensor Data Fusion. *Sensors* **2017**, *17*, 928.
21. Zhang, L.; Yang, F.B.; Ji, L.; Yuan, H.; Dong, A. A categorization method of infrared polarization and intensity image fusion algorithm based on the transfer ability of difference features. *Infrared Phys. Technol.* **2016**, *79*, 91–100.
22. Wang, Z.; Yang, F.; Peng, Z.; Chen, L.; Ji, L. Multi-sensor image enhanced fusion algorithm based on NSST and top-hat transformation. *Optik Int. J. Light Electron Opt.* **2015**, *126*, 4184–4190.
23. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353.
24. Li, D.W.; Yang, F.; Wang, X.X. Crop region extraction of remote sensing images based on fuzzy ARTMAP and adaptive boost. *J. Intell. Fuzzy Syst.* **2015**, *29*, 2787–2794.
25. Dempster, A.P. Upper and Lower Probabilities Induced by a Multi-valued Mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339.
26. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
27. Deng, Y. Generalized evidence theory. *Appl. Intell.* **2015**, *43*, 530–543.
28. Zhang, X.; Mahadevan, S. A Bio-Inspired Approach to Traffic Network Equilibrium Assignment Problem. *IEEE Trans. Cybern.* **2017**, *PP*, 1–12.
29. Guo, H.; Huang, H.; Liang, L. Fuzzy-robust design optimization with multi-quality characteristics. *Eng. Optim.* **2010**, *42*, 87–99.
30. Sun, S.; Tian, T.; Lin, H. State estimators for systems with random parameter matrices, stochastic nonlinearities, fading measurements and correlated noises. *Inf. Sci.* **2017**, *397*, 118–136.
31. Miao, B.; Dou, C.; Jin, X. Main Trend Extraction Based on Irregular Sampling Estimation and Its Application in Storage Volume of Internet Data Center. *Comput. Intell. Neurosci.* **2016**, *2016*, 9328062.
32. Jin, X.B.; Du, J.J.; Bao, J. Target Tracking of a Linear Time Invariant System Under Irregular Sampling. *Int. J. Adv. Rob. Syst.* **2012**, *9*, 219.
33. Yuan, K.; Xiao, F.; Fei, L.; Kang, B.; Deng, Y. Modeling Sensor Reliability in Fault Diagnosis Based on Evidence Theory. *Sensors* **2016**, *16*, 113.
34. Liu, Z.G.; Pan, Q.; Dezert, J.; Mercier, G. Credal classification rule for uncertain data based on belief functions. *Pattern Recognit.* **2014**, *47*, 2532–2541.
35. Han, D.; Liu, W.; Dezert, J.; Yang, Y. A novel approach to pre-extracting support vectors based on the theory of belief functions. *Knowl.-Based Syst.* **2016**, *110*, 210–223.
36. Liu, Z.G.; Pan, Q.; Dezert, J.; Martin, A. Adaptive imputation of missing values for incomplete pattern classification. *Pattern Recognit.* **2016**, *52*, 85–95.
37. Zhou, K.; Martin, A.; Pan, Q.; Liu, Z. ECMdd: Evidential c-medoids clustering with multiple prototypes. *Pattern Recognit.* **2016**, *60*, 239–257.
38. Ghorabae, M.K.; Zavadskas, E.K.; Amiri, M.; Antucheviciene, J. A New Method of Assessment Based on Fuzzy Ranking and Aggregated Weights (AFRAW) for MCDM Problems under Type-2 Fuzzy Environment. *Econ. Comput. Econ. Cybern Stud. Res.* **2016**, *50*, 39–68.
39. Chin, K.S.; Fu, C.; Wang, Y. A method of determining attribute weights in evidential reasoning approach based on incompatibility among attributes. *Computer. Ind. Eng.* **2015**, *87*, 150–162.
40. Fu, C.; Wang, Y. An interval difference based evidential reasoning approach with unknown attribute weights and utilities of assessment grades. *Computer. Ind. Eng.* **2015**, *81*, 109–117.
41. Fu, C.; Yang, J.B.; Yang, S.L. A group evidential reasoning approach based on expert reliability. *Eur. J. Oper. Res.* **2015**, *246*, 886–893.
42. Zhang, X.; Mahadevan, S.; Deng, X. Reliability analysis with linguistic data: An evidential network approach. *Reliab. Eng. Syst. Saf.* **2017**, *162*, 111–121.
43. Su, X.; Mahadevan, S.; Xu, P.; Deng, Y. Dependence Assessment in Human Reliability Analysis Using Evidence Theory and AHP. *Risk Anal.* **2015**, *35*, 1296–1316.

44. Zhou, D.; Tang, Y.; Jiang, W. A modified model of failure mode and effects analysis based on generalized evidence theory. *Math. Probl. Eng.* **2016**, *2016*, 4512383.
45. Yager, R.R.; Filev, D.P. Including probabilistic uncertainty in fuzzy logic controller modeling using Dempster–Shafer theory. *IEEE Trans. Syst. Man Cybern.* **1995**, *25*, 1221–1230.
46. Tang, Y.; Zhou, D.; Jiang, W. A New Fuzzy-Evidential Controller for Stabilization of the Planar Inverted Pendulum System. *PLoS ONE* **2016**, *11*, e0160416.
47. Zhou, K.; Martin, A.; Pan, Q.; Liu, Z.G. Median evidential c-means algorithm and its application to community detection. *Knowl.-Based Syst.* **2015**, *74*, 69–88.
48. Ma, J.; Liu, W.; Miller, P.; Zhou, H. An Evidential Fusion Approach for Gender Profiling. *Inf. Sci.* **2015**, *333*, 10–20.
49. Xu, P.; Su, X.; Mahadevan, S.; Li, C.; Deng, Y. A non-parametric method to determine basic probability assignment for classification problems. *Appl. Intell.* **2014**, *41*, 681–693.
50. Smets, P. Data fusion in the transferable belief model. *Inf. Fusion* **2000**, *1*, 21–33.
51. Lefevre, E.; Colot, O.; Vannoorenberghe, P. Belief functions combination and conflict management. *Inf. Fusion* **2002**, *3*, 149–162.
52. Tang, Y.; Zhou, D.; He, Z.; Xu, S. An improved belief entropy-based uncertainty management approach for sensor data fusion. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, doi:10.1177/1550147717718497.
53. Yager, R.R. On the Dempster–Shafer framework and new combination rules. *Inf. Sci.* **1987**, *41*, 93–137.
54. Murphy, C.K. Combining belief functions when evidence conflicts. *Decis. Support Syst.* **2000**, *29*, 1–9.
55. Deng, Y.; Shi, W.; Zhu, Z.; Liu, Q. Combining belief functions based on distance of evidence. *Decis. Support Syst.* **2004**, *38*, 489–493.
56. Chin, K.S.; Fu, C. Weighted cautious conjunctive rule for belief functions combination. *Inf. Sci.* **2015**, *325*, 70–86.
57. Jiang, W.; Zhuang, M.; Qin, X.; Tang, Y. Conflicting evidence combination based on uncertainty measure and distance of evidence. *SpringerPlus* **2016**, *5*, 1217.
58. Jousselme, A.L.; Grenier, D.; Bosse, E. A new distance between two bodies of evidence. *Inf. Fusion* **2001**, *2*, 91–101.
59. Shannon, C. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423.
60. Deng, Y. Deng entropy. *Chaos Solitons Fractals* **2016**, *91*, 549–553.
61. Yager, R.R.; Filev, D.P. Induced ordered weighted averaging operators. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1999**, *29*, 141–150.
62. O’Hagan, M. Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic. *Sign. Syst. Comput.* **1988**, *29*, 681–689.
63. Zeng, S.; Merigo, J.M.; Su, W. The uncertain probabilistic owa distance operator and its application in group decision making. *Appl. Math. Modell.* **2013**, *37*, 6266–6275.
64. Cho, S.B. Fuzzy aggregation of modular neural networks with ordered weighted averaging operators. *Int. J. Approx. Reason.* **1995**, *13*, 359–375.
65. Peng, Y.; Zhang, Y.; Tang, Y.; Li, S. An incident information management framework based on data integration, data mining, and multi-criteria decision making. *Decis. Support Syst.* **2011**, *51*, 316–327.
66. Fan, X.; Zuo, M.J. Fault diagnosis of machines based on D-S evidence theory. Part 1: D–S evidence theory and its improvement. *Pattern Recognit. Lett.* **2006**, *27*, 366–376.
67. Henry, D.; Cieslak, J.; Zolghadri, A.; Efimov, D. $H_\infty/H - LPV$ solutions for fault detection of aircraft actuator faults: Bridging the gap between theory and practice. *Int. J. Robust Nonlinear Control* **2015**, *25*, 649–672.
68. Cieslak, J.; Efimov, D.; Zolghadri, A.; Henry, D.; Goupil, P. Design of a non-homogeneous differentiator for actuator oscillatory failure case reconstruction in noisy environment. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2015**, *229*, 266–275.



Article

Complete Systematic Error Model of SSR for Sensor Registration in ATC Surveillance Networks

Ángel J. Jarama * , Jaime López-Araquistain, Gonzalo de Miguel and Juan A. Besada

Signals, Systems and Radiocommunications Department, Universidad Politécnica de Madrid, 28040 Madrid, Spain; jaime.lopez@grpss.ssr.upm.es (J.L.-A.); gonzalo@grpss.ssr.upm.es (G.d.M.); besada@grpss.ssr.upm.es (J.A.B.)

* Correspondence: ajarama@grpss.ssr.upm.es

Received: 14 July 2017; Accepted: 19 September 2017; Published: 21 September 2017

Abstract: In this paper, a complete and rigorous mathematical model for secondary surveillance radar systematic errors (biases) is developed. The model takes into account the physical effects systematically affecting the measurement processes. The azimuth biases are calculated from the physical error of the antenna calibration and the errors of the angle determination dispositive. Distance bias is calculated from the delay of the signal produced by the refractivity index of the atmosphere, and from clock errors, while the altitude bias is calculated taking into account the atmosphere conditions (pressure and temperature). It will be shown, using simulated and real data, that adapting a classical bias estimation process to use the complete parametrized model results in improved accuracy in the bias estimation.

Keywords: bias estimation; radar modeling; measurement error; air traffic control; sensor fusion

1. Introduction

Nowadays radars are the basic element in Air Traffic Control (ATC) networks. Modern sensors as Wide Area Multilateration (WAM) [1] and Automatic Dependent Surveillance-Broadcast (ADS-B) [2] have higher performance than radars with lower cost, so in the future it's expected they will partially take the place of radars. In any case, radars will still be used as a backup network for many years. The type of radar most used in ATC sensor network is the Secondary Surveillance Radar (SSR), being Mode S radar an enhanced version of this system [3]. SSR is a rotating 2D radar sending an interrogation through an antenna with high directivity in azimuth and low directivity in elevation (2D antenna). Afterwards the aircraft transponder replies with an answer codifying barometric altitude (Mode C) and identification (Mode S and/or Mode A) data. SSR radars estimate the position of the aircraft in polar coordinates, using the distance, the azimuth (both measured by the radar) and the altitude (measured by an airborne barometric altimeter).

To enable surveillance of wide areas radars with overlapped coverages are deployed. Therefore, aircraft measures (plots) from different sensors are sent to a control center where the data from all sensors is fused in order to get a unique trajectory estimation (track) for each aircraft. All the measures are transformed from radar polar local coordinates to a common coordinated system, usually Cartesian coordinates projected over the stereographic plane [4]. To make this data fusion process stable and accurate, the systematic errors affecting the measures from each radar must be corrected before the change of coordinates. If this correction isn't applied the measures of the same aircraft from different sensors are misaligned and the track could be either very unstable (apparent zig-zag maneuvers induced by measurements misalignment), or in extreme cases it could even be split into several tracks (i.e., one per sensor).

Radar data fusion systems usually have an algorithm that estimates the misalignment error using the measures from the current aircraft in the airspace (opportunity traffic). These algorithms are usually

based in the least squares (LS) method (or similar approaches, such as WLS, or MSE estimators [5]) and have the need of a parametrized mathematical model that links the systematic errors in azimuth, distance and altitude with the sensor physical non-idealities.

In the abundant literature about biases estimation methods in ATC networks such as [6–8], the mathematical bias models used are simple (modeling just a range and azimuth offset and a range bias term proportional to range) because the objective of the papers was showing the estimation methods. Recently in [9] a novel registration algorithm has been presented where the track state and sensor biases are estimated simultaneously modeling the sensors with range and azimuth offset. Other related examples can be found in [10,11]. In real applications, these models are too simple, resulting in reduced performance when used with real data. At the same time, the development of mathematical models for biases is a classic topic for radio electric signals. For instance, in [12] a model is developed for measurement errors in azimuth for antennas in Deep Space Networks. Meanwhile, [13–17] it's developed an azimuth bias model for 3D antennas. Due to the dispersion of the literature about the systematic error modelling, it is apparent quite often both in the literature and in real systems there is much effort on improving estimation processes but little effort is given to rigorous modelling of the error sources.

The objective of this paper is to derive a complete systematic error model for ATC radars enabling an important improvement of the bias estimation processes. In Section 2 we will introduce a measurement model for azimuth, distance and altitude measurements. Then, in Sections 3–5, we will derive the relation between the measurement error and the measurement biases terms. Section 6 describes a simple bias estimation process (based on opportunity traffic) to be used to show the improvement in estimation due to improved error modelling. Then, Section 7 includes simulation and real data results showing the aforementioned improvement. In the results it is shown the improvement of the alignment that can be obtained using a complete model in comparison with a simple model. Finally, Section 8 concludes the paper and gives some clues about future research.

2. Measurement Model

Radars used in ATC typically use a two dimensional rotating antenna and have only the capability to measure aircraft range and azimuth. The azimuth of the target is the angle of the antenna boresight while the target altitude is measured by the aircraft navigation system and can be communicated to radar through a data link. Primary Surveillance Radars (PSR) only can determine the range and the azimuth of the target due to the lack of data link capability. Nowadays in ATC systems primary radars are used as backup system to secondary radar network. SSR is based in the same concept that PSR but in this case, the SSR has data link capability. SSR sends an interrogation message and the aircraft responds with its Mode A/Mode S (identifying code) or mode C (barometric altitude).

The azimuth and the range of the target are determined respectively with antenna boresight (usually corrected using monopulse [3]) and the time lapse between the radar interrogation and its response. Range and azimuth are typically referenced to a local Cartesian coordinated system: y -axis pointing to north, x -axis pointing to East and z -axis pointing to up (ENU system). Azimuth reference has its origin in the y -axis and it grows in clockwise. Barometric altitude is measured by an airborne barometric altimeter with reference at mean sea level. The relation between radar coordinates (ρ_m, θ_m, h_m) and local Cartesian coordinates (x_m, y_m) is:

$$\begin{aligned} x_m &= \rho_m \sin(\theta_m) \cos(\varphi_m) \\ y_m &= \rho_m \cos(\theta_m) \cos(\varphi_m) \end{aligned} \quad (1)$$

where ρ_m, θ_m, h_m are the range, azimuth and altitude of the measurement and φ_m is the target elevation with respect to i -th sensor horizontal plane, which can be calculated as:

$$\varphi_m = \arcsin\left(\frac{2R(h_m - h_i) + h_m^2 - h_i^2 - \rho_m^2}{2\rho_m(R + h_i)}\right) \quad (2)$$

being h_i the geometric altitude of the i -th sensor with reference at mean sea level. R is the local Earth radius at the radar position, calculated as described in [18]. Local Cartesian coordinates will be used as middle step in the conversion between radar measures and common tracking coordinates for all sensors.

The radar measurement model can be summarized as:

$$\begin{aligned}\rho_m &= \rho + n_\rho + \Delta\rho \\ \theta_m &= \theta + n_\theta + \Delta\theta \\ h_m &= h + n_h + \Delta h\end{aligned}\quad (3)$$

where n_ρ , n_θ and n_h are the noise measurement errors for range, azimuth and altitude respectively, $\Delta\rho$, $\Delta\theta$ and Δh are systematic error terms due to bad system calibration, and ρ , θ and h are the ideal range, azimuth and altitude of the target. Noise terms are modelled typically as uncorrelated additive white noises (usually considered Gaussian distributed), while systematic error terms can be considered constants or slow time variant (bias terms). Bias terms of range, azimuth and altitude depend on target position and will be deduced in Sections 3–5 for SSR sensors. They are composed by several environmental and equipment perturbations that are common to all the measurements of each sensor. The parameters of the mathematical model may change with time, due to variable weather conditions and hardware aging.

In surveillance data fusion systems the measures obtained from different kind of sensor are fused in order to improve the information update ratio and the accuracy of the aircraft trajectory estimation. At the moment of the data fusion all measurement must be expressed in a common coordinated reference system. There are several coordinated systems that can be used in ATC. For small scenarios, flat Earth model can be assumed, therefore Cartesian coordinates are precise enough to track the trajectories with several sensors. When the scenario is larger and sensors are widely separated, the Earth curvature must be considered and a different coordinate system must be used. A typical solution is to use stereographic projection to project all measures expressed in local Cartesian reference systems to the same plane [4]. Afterwards, the multisensor tracking process will be performed in the stereographic reference system for horizontal track. Aircraft barometric altitude is usually tracked separately in ATC applications.

3. Azimuth Bias

Azimuth bias can be modelled as the superposition of several terms that can be separated in three different groups. The first one is the erroneous orientation of the antenna boresight due to misalignments between the antenna and the rotation axis. The second one is the non-orthogonality between the rotation axis and the Earth surface. And the last one is the calibration error of the dispositive of angle determination (azimuth encoder). Some of these terms will have the same effect in the measures and they cannot be distinguished in the estimation process using measures from radar. For example, the azimuth offset of the antenna, the azimuth offset of the rotation axis and the azimuth offset of the encoder will have equivalent effects and only one of them will be considered. In this case, the model adds all the effects in a single parameter.

In other cases, the superposition of these terms will depend on aircraft position. Antenna skew and rotation axis skew will be distinguishable if the aircraft are distributed in azimuth. For these two bias terms both the elevation and azimuth diversity of the aircraft are needed in order to estimate them. Generally, the aircraft will be distributed in the airspace (position and altitude) but sometimes there will be cases where all the aircraft will be concentrated in a small rank of azimuth, altitude or range. In these cases, the bias estimation algorithms will not have the capability to separate the different azimuth and bias components (there would be an observability problem [15]).

3.1. Antenna Deviation

The first factor that affects the azimuth bias is the deviation between the vertical axis of the antenna reference and the reference of the rotation axis. This deviation will be defined with three parameters describing the rotation in the three axes of the Cartesian coordinate system. In order to determine the antenna deviation, the coordinated system is defined as: y -axis, parallel to the ideal antenna boresight; z -axis, orthogonal to the y -axis, resulting that both axes compose the ideal plane of the 2D antenna pattern (z -axis is the ideal antenna rotation axis); the x -axis, orthogonal to the y -axis and z -axis positive to the right.

In order to derive the model, the biased antenna coordinated system (x', y', z') is calculated. The axes of the coordinated system (x, y, z) are rotated to get the new system (x', y', z') and the measured azimuth (θ_m) is calculated in the new system. Under the assumption that the deviation angles of the antenna axes are small (below 1°), the order of the rotation around each axis has a negligible influence in the model. The three rotation angles (Figure 1) correspond to the following bias parameters components:

- Azimuth offset (θ_0): it is generated with the rotation of the antenna in the horizontal plane. This offset is constant for all the targets of the sensor. This rotation is defined positive in counter-clockwise direction.
- Antenna squint (s_{ant}): it is the bias produced due to the rotation in the orthogonal plane to the antenna boresight. This component produces an azimuth bias dependent on the target elevation (angle of line of sight respect to horizontal plane). This rotation is defined positive in counter-clockwise direction.
- Antenna tilt (t_{ant}): produced due to the rotation of the plane of the 2D antenna pattern around the x -axis. As radars used in ATC applications do not measure the elevation, this bias has not consequences in the first order terms of the azimuth bias. This rotation is defined positive in clockwise direction.

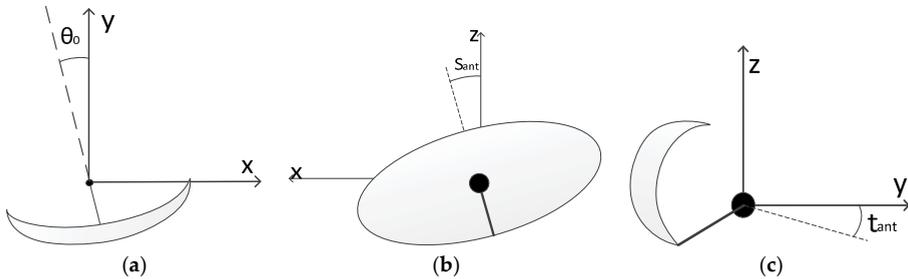


Figure 1. Representation of: (a) antenna azimuth offset, (b) antenna squint (c) and antenna tilt.

The rotation matrix for each one of the axes are:

$$R_z(\theta_0) = \begin{pmatrix} \cos(\theta_0) & \sin(\theta_0) & 0 \\ -\sin(\theta_0) & \cos(\theta_0) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$R_x(t_{ant}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(t_{ant}) & -\sin(t_{ant}) \\ 0 & \sin(t_{ant}) & \cos(t_{ant}) \end{pmatrix} \quad (5)$$

$$R_y(s_{ant}) = \begin{pmatrix} \cos(s_{ant}) & 0 & -\sin(s_{ant}) \\ 0 & 1 & 0 \\ \sin(s_{ant}) & 0 & \cos(s_{ant}) \end{pmatrix} \quad (6)$$

Applying these rotations, the resultant coordinate system is:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = R_z(\theta_0) \cdot R_x(t_{ant}) \cdot R_y(s_{ant}) \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (7)$$

Due to the convention in geographical systems of the azimuth with the origin in the y -axis, positive in clockwise, the measured azimuth ($\theta_{m,ant}$) is:

$$\theta_{m,ant} = \arctan\left(\frac{x'}{y'}\right) \quad (8)$$

These calculations are made under two assumptions: the error is only caused by the deviations of the antenna and this antenna can only acquire the targets that are in the 2D antenna main lobe. A multidimensional Taylor series on the bias variables θ_0 , s_{ant} and t_{ant} is made to derive the following linearized approximation:

$$\begin{aligned} \theta_{m,ant} &\approx \theta + \theta_0 - s_{ant} \tan(\varphi) \cos(\theta) + t_{ant} \tan(\varphi) \sin(\theta) \\ &+ \left(\frac{1}{2} \sin(\theta) \cos(\theta) - \frac{\sin(\theta) \cos(\theta)}{\cos^2(\varphi)}\right) s_{ant}^2 - \left(\frac{1}{2} \sin(\theta) \cos(\theta) - \frac{\sin(\theta) \cos(\theta)}{\cos^2(\varphi)}\right) t_{ant}^2 \\ &+ \left(\cos^2(\theta) - \frac{2 \cos^2(\theta)}{\cos^2(\varphi)} + \frac{1}{\cos^2(\varphi)}\right) t_{ant} s_{ant} + \dots \end{aligned} \quad (9)$$

where φ is the ideal elevation of the aircraft. The variable θ may be particularized to zero in this case as the antenna only measures the aircraft that are into the antenna main lobe ($\theta \approx 0$). Assuming that θ_0 , s_{ant} and t_{ant} are small values then the higher order terms may be assumed to be negligible, resulting in:

$$\theta_{m,ant} \approx \theta_0 - s_{ant} \tan(\varphi) \quad (10)$$

The previous measure just contains the projection of the antenna deviation parameters in the azimuth component ($\theta \approx 0$; noise and other bias terms are neglected for this calculation), resulting in the following additive azimuth bias term:

$$\Delta\theta_{ant} = \theta_0 - s_{ant} \tan(\varphi) \quad (11)$$

3.2. Axis Skew

The axis skew is the inclination of the rotation axis. It causes an effect similar to the antenna deviation but in this case the bias also depends on the azimuth. For this model we use the local Cartesian coordinated system in the radar position: the y -axis pointing to the north and the x -axis pointing to the east both in the Earth horizontal plane. As in the previous subsection the biases of the azimuth can be calculated with rotations of the system axis as is represented in Figure 2.

In order to be coherent with the names of the rotations, the rotation in y -axis will be named axis squint and the rotation in x -axis will be named axis tilt. In this case, the tilt gives errors on the measures because it is the rotation in the x -axis and not in the orthogonal plane to the antenna boresight.

In this case the rotation matrix are:

$$R_x(t_{axis}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(t_{axis}) & -\sin(t_{axis}) \\ 0 & \sin(t_{axis}) & \cos(t_{axis}) \end{pmatrix} \quad (12)$$

$$R_y(s_{axis}) = \begin{pmatrix} \cos(s_{axis}) & 0 & -\sin(s_{axis}) \\ 0 & 1 & 0 \\ \sin(s_{axis}) & 0 & \cos(s_{axis}) \end{pmatrix} \tag{13}$$

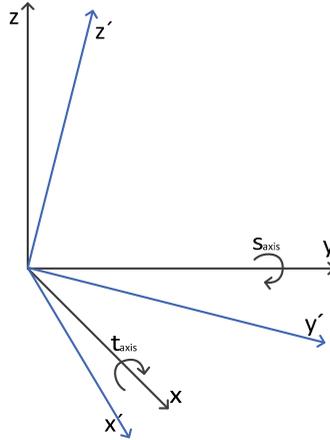


Figure 2. Skew of the rotation axis of the radar.

The new coordinates system (x', y', z') is:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = R_x(t_{axis}) \cdot R_y(s_{axis}) \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{14}$$

The measured azimuth is calculated again ($\theta_{m,axis} = \arctan(\frac{x'}{y'})$) and the result is approximated linearizing with a multidimensional Taylor series on the bias variables s_{axis} and t_{axis} . In this case the azimuth is not particularized to zero ($\theta \approx 0$) because the reference axes are fixed to north and the ideal measurement can have any value of azimuth. The linear approximation, rejecting higher order terms in Taylor series is:

$$\theta_{m,axis} \approx \theta + t_{axis} \tan(\varphi) \sin(\theta) - s_{axis} \tan(\varphi) \cos(\theta) \tag{15}$$

Therefore, the azimuth bias due to the axis skew is:

$$\Delta\theta_{axis} = \theta_{m,axis} - \theta \approx t_{axis} \tan(\varphi) \sin(\theta) - s_{axis} \tan(\varphi) \cos(\theta) \tag{16}$$

An alternative equivalent model to the one in (16) can be derived as follows. Under the assumption that the inclination of the axis is small, the spherical triangles with components t_{axis} and s_{axis} can be approximated by the right triangle of Figure 3, expressed in terms of β_{axis} and α_{axis} where β_{axis} represents the total skew and α_{axis} the direction where the axis is skewed. The new relation is:

$$\begin{aligned} s_{axis} &= \beta_{axis} \sin(\alpha_{axis}) \\ t_{axis} &= \beta_{axis} \cos(\alpha_{axis}) \end{aligned} \tag{17}$$

Substituting (17) in (16) the final model could be written, after some minor algebra, as:

$$\Delta\theta_{axis} = \beta_{axis} \sin(\theta - \alpha_{axis}) \tan(\varphi) \tag{18}$$

The azimuth offset due to this effect at the direction of the skew is zero, and it follows a sinusoidal law with azimuth. The model with s_{axis} and t_{axis} parameters in (16) is equivalent to the model with β_{axis} and α_{axis} parameters in (18). The former model (16) results generally in better stability when used for the development for bias estimation algorithms.

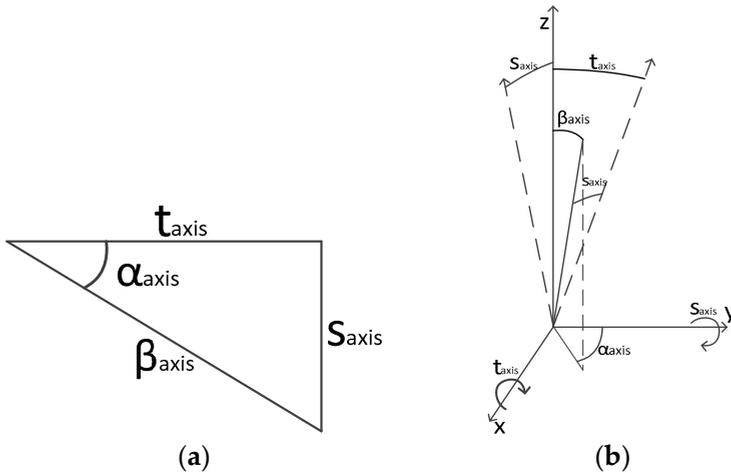


Figure 3. (a) Right triangle of the axis skew and (b) relation between the rotation of the axis and the skew of the rotation axis and its direction.

3.3. Optical Encoder

In modern radars the azimuth of the antenna boresight is determined with an optical encoder that is allocated in the rotating radar shaft. In the installation of this encoder, small calibration errors can be made and these are transformed in azimuth biases. One of the errors is the misalignment between azimuth reference of the encoder and the axis, this is an azimuth offset. This error is the same kind that the azimuth offset of the antenna and the azimuth offset of the rotating axis, described in Section 3.1.

Other error to be considered is the eccentricity bias. Usually this is the encoder error with the biggest impact in the measures [19]. In the installation, the rotation axis (O) is not exactly placed in the geometric center (O') of the encoder.

The resultant geometry is shown in the Figure 4, supposing that the difference of the centers is in the azimuth $\theta = 0$. Resolving the triangle shown in Figure 5 and making a deduction similar to [20,21] we have the following relation:

$$\left. \begin{aligned} A &= \Delta R \sin(\theta) \\ A &= R \sin(\alpha) \end{aligned} \right\} \Rightarrow \sin(\alpha) = \frac{\Delta R}{R} \sin(\theta) \tag{19}$$

where, ΔR is the eccentricity offset, R is radius of the encoder and θ and α are angles of the triangle shown in Figure 5.

Using the relation between sine and cosine for α ($\cos(\alpha) = \sqrt{1 - \sin^2(\alpha)}$) we also have the following relation:

$$\cos(\alpha) = \sqrt{1 - \left(\frac{\Delta R}{R}\right)^2 \sin^2(\theta)} = \frac{1}{R} \sqrt{1 - \Delta R^2 \sin^2(\theta)} \tag{20}$$

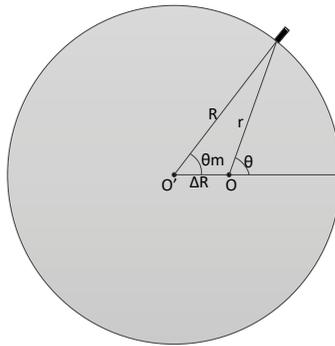


Figure 4. Optical encoder with eccentricity bias.

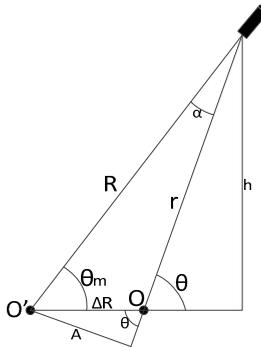


Figure 5. Resolution of the resultant triangle for eccentricity bias.

On the other hand, from Figure 5 we can also see:

$$r = R \cos(\alpha) - \Delta R \cos(\theta) \Rightarrow r = -\Delta R \cos \theta + \sqrt{R^2 - \Delta R^2 \sin^2(\theta)} \tag{21}$$

Finally:

$$\left. \begin{aligned} h &= R \sin(\theta_m) \\ h &= r \sin(\theta) \end{aligned} \right\} \Rightarrow \theta_m = \arcsin\left(\frac{r}{R} \sin(\theta)\right) \tag{22}$$

The measured azimuth is also linearized with a Taylor series on the bias variable ΔR in order to get an approximation to model. Assuming that $\frac{\Delta R}{R} \ll 1$ the approximation of $\left(\frac{\Delta R}{R}\right)^2 = 0$ can be made. With these simplifications, the azimuth measurement can be approximated as:

$$\theta_m \approx \theta - \frac{\Delta R}{R} \sin(\theta) \tag{23}$$

The second error produced by the optical encoder is due to the swash between the encoder and the horizontal plane. A small swash of the encoder produces that the projection of the circumference is an ellipse and the determination of the azimuth has errors when this azimuth does not point to the ellipse axis.

In Figure 6a a schematic model of the optical encoder is represented. The circle line on the $x-y'$ plane represents the mechanical encoder and the dotted line on the $x-y$ plane represents the projection of the rotated mechanical encoder circumference on the horizontal plane. The ideal azimuth

is represented on the horizontal plane but the measured azimuth is obtained by the rotated encoder. In Figure 6b both mechanical encoder circumference and its projection on the horizontal plane are represented in the same plane in order to represent the 3D projection in a 2D figure to see the difference between the real azimuth and the measured azimuth. Following the methods used in [18] to convert geocentric latitude in reduced latitude we can obtain the relation between θ_m and θ . Using the next variables:

$$\begin{aligned} x &= r \sin(\theta_m) \\ y_m &= r \cos(\theta_m) \\ y &= y_m \cos(s_{enc}) \end{aligned} \tag{24}$$

and solving the triangle for θ :

$$\theta = \operatorname{atan}\left(\frac{x}{y}\right) = \operatorname{atan}\left(\frac{\tan(\theta_m)}{\cos(s_{enc})}\right) \tag{25}$$

and then θ_m can be calculated as:

$$\theta_m = \operatorname{atan}(\cos(s_{enc}) \tan(\theta)) = \operatorname{atan}\left(\cos(s_{enc}) \frac{x}{y}\right) \tag{26}$$

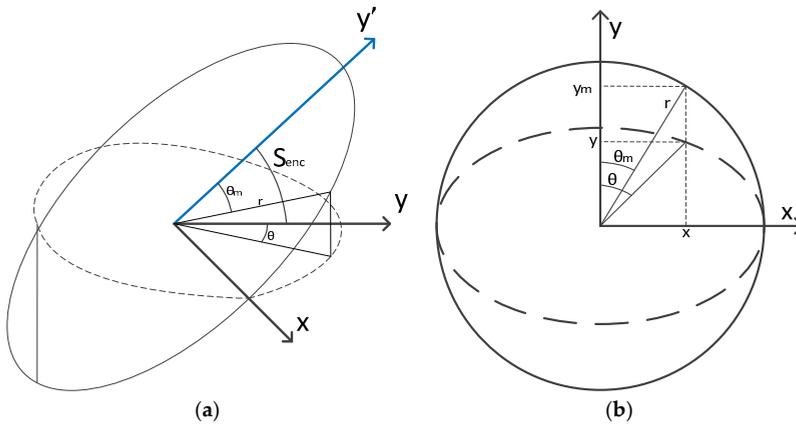


Figure 6. Swash of the optical encoder. (a) Schematic model of the optical encoder; (b) Mechanical encoder circumference and its projection on the horizontal plane.

A Taylor series of order 2 on the bias variable s_{enc} is made to get an approximation of the measure:

$$\theta_m \approx \theta - \frac{1}{2} \cos(\theta) \sin(\theta) s_{enc}^2 = \theta - \frac{s_{enc}^2}{4} \sin(2\theta) \tag{27}$$

In this Taylor series, the first order term is zero and the second order term is considered. As this term is raised to second power, with small values of s_{enc} the value of the bias will be negligible. In both terms related with the encoder the error has been calculated supposing that the difference of the centers and the swash are in the azimuth $\theta = 0$. Supposing that in reality the azimuth of the difference of centers is α_{ecc} and the azimuth of the swash is α_{enc} the resultant encoder bias, comprising both the effects described in (23) and (27), is:

$$\Delta\theta_{enc} = -\frac{s_{enc}^2}{4} \sin(2\theta - 2\alpha_{enc}) - \frac{\Delta R}{R} \sin(\theta - \alpha_{ecc}) \tag{28}$$

An equivalent model can be derived by using trigonometric relations, in order to express all the biases with a linear structure similar to the one in Equation (16):

$$\Delta\theta_{enc} = s_{enc,s} \sin(2\theta) + s_{enc,c} \cos(2\theta) + \Delta R_y \sin(\theta) - \Delta R_x \cos(\theta) \quad (29)$$

where:

$$s_{enc,s} = \frac{s_{enc}^2 \sin^2(\alpha_{enc}) - s_{enc}^2 \cos^2(\alpha_{enc})}{4} \quad (30)$$

$$s_{enc,c} = s_{enc}^2 \cos(\alpha_{enc}) \sin(\alpha_{enc})$$

and ΔR_y and ΔR_x are the terms of the center deviation in the y -axis and in the x -axis.

3.4. Azimuth Bias Composition

The global azimuth bias model is a composition of the three bias elements previously discussed. Neglecting noise effects, in a radar, the measured azimuth of the antenna is the azimuth where the rotation axis is pointing plus the antenna bias. The azimuth of the rotation axis is the azimuth indicated by the encoder plus the axis bias. Finally, the azimuth indicated by the encoder is the ideal azimuth plus the encoder bias. With this:

$$\begin{aligned} \theta_{enc} &= \theta + \Delta\theta_{enc} \\ \theta_{axis} &= \theta_{enc} + \Delta\theta_{axis} \\ \theta_m &= \theta_{ant} = \theta_{axis} + \Delta\theta_{ant} \end{aligned} \quad (31)$$

and then:

$$\theta_m = \theta + \Delta\theta = \theta + \Delta\theta_{ant} + \Delta\theta_{axis} + \Delta\theta_{enc} \quad (32)$$

With this, the complete model for azimuth bias is:

$$\begin{aligned} \Delta\theta &= \Delta\theta_{ant} + \Delta\theta_{axis} + \Delta\theta_{enc} \\ &= \theta_0 - s_{ant} \tan(\varphi) + \beta_{axis} \sin(\theta - \alpha_{axis}) \tan(\varphi) - \frac{1}{4} \sin(2\theta - 2\alpha_{enc}) s_{enc}^2 - \frac{\Delta R}{R} \sin(\theta - \alpha_{exc}) \end{aligned} \quad (33)$$

This model should not be used in general in the bias estimation algorithms due to instability problems due to lack of linearity of some of part of the model, although its physical meaning is clear. The alternative equivalent model (preferred for bias estimation due to its linearity with respect to bias parameters) should be obtained composing (11), (16), (29) and (32), resulting:

$$\begin{aligned} \Delta\theta &= \theta_0 - s_{ant} \tan(\varphi) + [t_{axis} \sin(\theta) - s_{axis} \cos(\theta)] \tan(\varphi) + s_{enc,s} \sin(2\theta) \\ &\quad + s_{enc,c} \cos(2\theta) + \Delta R_y \sin(\theta) - \Delta R_x \cos(\theta) \end{aligned} \quad (34)$$

4. Range Bias

Range biases are again due to several factors. The main factors that compose the biases are:

- Transponder delay: every aircraft has a different transponder delay (typically between -75 m and 75 m, usually assumed to be uniformly distributed). This aircraft dependent bias must be estimated for each aircraft in the tracking phase. The range model developed in this paper is for biases due to sensor and environment. As the estimation will be made with many targets, the mean of the delays of all the aircraft tends to be near zero. For more details on this bias term and methods to estimate it, see [8].
- Temporal reference error: radars measure the difference of time between the interrogation and the arrival of the response. Usually there is a small constant error in the reference time and it produces a constant error (ρ_0) in the range estimation.

- Clock error: A small error in the clock calibration may produce a linear term in the range estimation [22]. In modern radar this term can be considered negligible due to the precision of the clocks.
- Propagation error: The conversion between time and distance is made using the speed of light in the ISA (International Standard Atmosphere) at mean sea level. However, the speed of light changes along the propagation path due to the change in the refraction index with altitude and weather conditions. Any change in the real speed of light will produce a range bias term ($\Delta\rho_{prop}$).

Considering the range biases terms previously explained, the range bias ($\Delta\rho$, as defined (3)) results:

$$\Delta\rho = \rho_0 + \Delta\rho_{prop} \tag{35}$$

In cases when the clock has a bad calibration, the bias has additionally a linear term dependent on the clock error, and parameterized through a range gain factor (α_{clk}), being the complete model:

$$\Delta\rho = \rho_0 + \alpha_{clk} \rho + \Delta\rho_{prop} \tag{36}$$

Next, we will focus on the propagation term.

4.1. Propagation Error

The index of refraction of the atmosphere changes with the altitude and this affects to the local speed of light. As the conversion between time and range is made with a constant speed of light, the determination of the position of the aircraft from the radar generally has biases (due to change of speed of light with altitude and weather conditions).

As the bias is produced by accumulated error in the speed of light along the propagation path, the range bias will be a function of the target range. In order to model the relation between the ideal range and the bias we will use a spherically stratified atmosphere. The exponential model of the refractivity [23] is used because it models the typical atmosphere variations in the rank of altitude used in ATC applications:

$$N(h) = N_s \exp\left(-\frac{h}{H}\right) \tag{37}$$

where $N(h)$ is the refractivity at a given altitude (h) over mean sea level, H is a reference altitude (equal to 6950 m), and $N(0) = N_s$ (with typical values between 300 and 350). The relation between the refractivity and the index of refraction ($n(h)$) is:

$$n(h) = 1 + N(h) \times 10^{-6} \tag{38}$$

The model in Equation (37) can be used also redefining altitude (h) to be referred to radar altitude, which just results in a change of the N_s constant, to be now the refractivity at the radar altitude over mean sea level. This will be the altitude reference and refractivity model to be used for the rest of this section.

The height gradient of the index of refraction bends the ray, resulting in a slightly increased distance from that of the straight line used to model propagation on free space. Specifically, as it is deduced in [24] using a ray tracing procedure with the Snell's law for spherically stratified media [25], the geometrical distance of the bent ray path may be calculated as:

$$s(h_t, \varphi_{0t}) = \int_0^{h_t} \frac{dh}{\sqrt{1 - \left\{ n_0 \cos(\varphi_{0t}) \left[\frac{n(h)}{1 + \frac{h}{R_e}} \right] \right\}^2}} \tag{39}$$

where φ_{0t} is the elevation ray angle observed by the radar, h_t is the target altitude relative to the radar, n_0 is the index of refraction at radar altitude, and R_e is the Earth radius plus radar height.

Additionally, the speed of light varies as it traverses different heights with different associated index of refraction. The effective distance (measured distance assuming constant speed of light, is affected by the propagation error), and can be calculated as proposed in [24]:

$$r(h_t, \varphi_{0t}) = \int_0^{h_t} \frac{n(h)dh}{\sqrt{1 - \left\{ n_0 \cos(\varphi_{0t}) \left[\frac{n(h)}{1 + \frac{h}{R_e}} \right] \right\}^2}} \tag{40}$$

In the following we will work with the approximation that the geometrical distance between the target and radar is almost equal to a straight-line (free-space) path length [24] ($\rho \approx s(h_t, \varphi_{0t})$). Then, the range bias due to propagation can be calculated as:

$$\Delta\rho(h_t, \varphi_{0t}) = r(h_t, \varphi_{0t}) - s(h_t, \varphi_{0t}) \tag{41}$$

Both distances (geometrical distance and effective distances), and their difference, can be numerically estimated for different values of h_t and φ_{0t} . With these pairs we can get a table of range biases ($\Delta\rho$) and geometrical distances (s). In these estimations we work with the approximation that the geometrical distance between the target and radar is equal to the ray path length [24]. Without this approximation the estimations are similar, but at distances lower than 20 Km the results of the biases are different. But in these places the total bias is very low (between 2 m and 6 m, depending on the altitude) and the approximation does not introduce significant error. The resultant range error is represented in the Figure 7, where each line represents the range error for a different altitude (h_t) as a function of slant range.

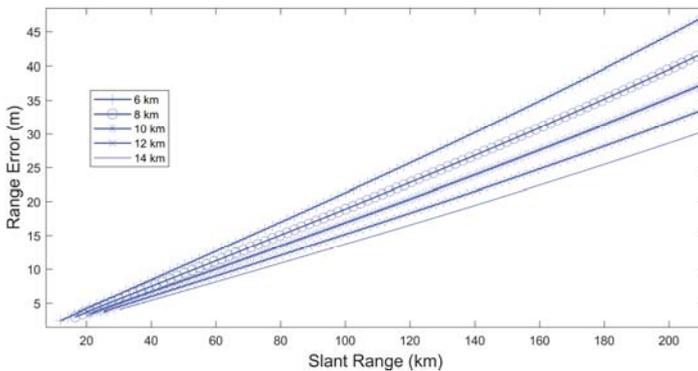


Figure 7. Range bias for aircraft at different altitudes.

As it is seen [26] two relevant characteristics can be observed. The first one is that the range error depends on the altitude of the aircraft. The second one is that a second-grade polynomial seems to be good enough to approximate the range error dependency with the slant range for every altitude [27]. Considering only this polynomial, the range error can be expressed as:

$$\Delta\rho(\rho, h) = \beta_1(h)\rho + \beta_2(h)\rho^2 \tag{42}$$

where $\beta_1(h)$ and $\beta_2(h)$ are polynomial coefficients, different for each altitude.

This model does not allow to make a good estimation of the parameters with opportunity traffic, because the number of aircraft in each altitude layer is reduced. We must find a model which allows the simultaneous use of aircraft measures at all altitudes in the estimation process.

A key idea in the derivation of such model is to search for decoupled range and altitude dependencies in $\Delta\rho(\rho, h)$:

$$\Delta\rho(\rho, h) = f(\rho) \cdot g(h) \tag{43}$$

To do so, the quotient between a parabola in an altitude and the parabola at an arbitrary reference altitude ($h_r = 14,000$ m in the following) for the same slant range is made:

$$g(\rho, h) = \frac{\Delta\rho(\rho, h)}{\Delta\rho(\rho, h_r)} \tag{44}$$

This quotient is shown as a function of the slant range in Figure 8 and can be observed that for each altitude it is almost constant. The right end of the horizontal lines in this figure corresponds to the radio-wave propagation horizon. This result demonstrates that the range bias at a specific altitude is the range bias at the reference altitude multiplied by a constant, supporting Equation (43), as the dependency of $g(\rho, h)$ with the slant-range is negligible (i.e., it could be substituted by $g(h)$), and $f(\rho)$ would be directly $\Delta\rho(\rho, h_r)$, a function not depending of h . From those ideas, we have that:

$$f(\rho) = \beta_1(h_r)\rho + \beta_2(h_r)\rho^2 = \alpha_1\rho + \alpha_2\rho^2 \tag{45}$$

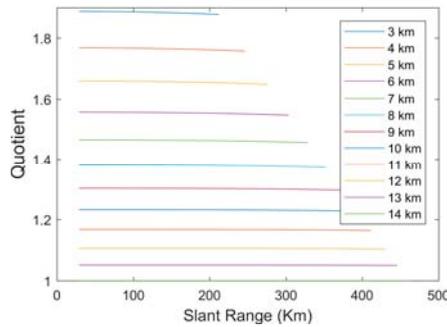


Figure 8. Quotient between range bias at an altitude and bias at 14,000 m for the same slant range.

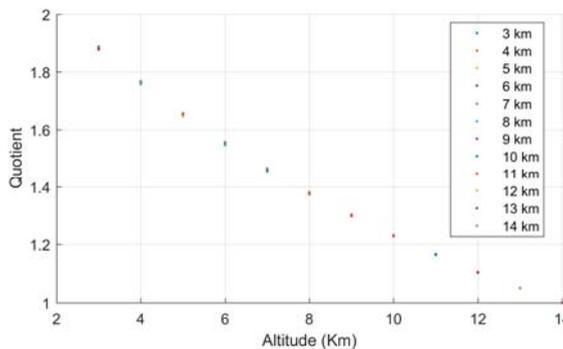


Figure 9. Ratio between range bias at an altitude and bias at reference altitude (14,000 m) for the same slant range as a function of altitude.

In order to derive $g(h)$ we have represented in the Figure 9 the quotient between the range bias at an altitude and the bias at a reference altitude as a function of altitude. In this figure, it can be seen

that a linear model is a good approximation. A second order polynomial would be an almost exact approximation but it implies the addition of another parameter that would make the convergence of the bias estimation processes slower with reduced accuracy gains.

Then $g(h)$ can be modelled with a linear function normalized to reference altitude (h_r):

$$g(h) = \frac{\Delta\rho(\rho, h)}{\Delta\rho(\rho, h_r)} \approx 1 + \alpha_3 \left(1 - \frac{h}{h_r}\right) \quad (46)$$

Finally, the contribution of the tropospheric propagation to the range bias can be modelled, combining Equations (43), (45) and (46):

$$\Delta\rho_{prop} = \left(\alpha_1\rho + \alpha_2\rho^2\right) \left[1 + \alpha_3 \left(1 - \frac{h}{h_r}\right)\right] \quad (47)$$

4.2. Range Error Composition

The final range bias is the addition on the one hand of the propagation bias due to the atmosphere and on the other hand to the constant bias produced by the transponder delay and the error of reference time:

$$\Delta\rho = \Delta\rho_0 + \left(\alpha_1\rho + \alpha_2\rho^2\right) \left[1 + \alpha_3 \left(1 - \frac{h_r}{h_r}\right)\right] \quad (48)$$

5. Altitude Bias

Civil surveillance radars used in ATC do not measure elevation of the targets. The reply of the aircraft transponder to the SSR interrogation codifies the altitude of the aircraft in the message (Mode C), completing the 3D position determination by the radar sensor. Aircraft altitudes may be of two different types:

- Geometric altitude: it is the Euclidean distance between the mean sea level (MSL) and the position of the aircraft.
- Barometric altitude: this altitude is calculated from the air pressure of the atmosphere at aircraft location. The conversion between pressure and altitude is made assuming International Standard Atmosphere (ISA) model (temperature 288.15 K and pressure 101,325 Pa, at MSL) [28]. Barometric altitude is the same that geometric altitude only when the atmospheric conditions are the ISA conditions.

Barometric altitude is used in ATC systems, because all aircraft must always measure the pressure to fly and its flight performance depends critically on this pressure, barometers are easy to be calibrated and vertical separation can be done easily using this magnitude. On board certified sensors for geometric altitude usable in all phases of flight (such as GPS) have only recently become widely available, and this information is not always available in the ground. All the surveillance systems send barometric altitude but only some of them can send the geometric altitude. Using the barometric altitude usually implies systematic errors in altitude determination that are almost equal for every aircraft and those depends on the pressure and temperature.

This altitude bias term, when the polar measure is transformed to the horizontal plane, introduces a systematic error in the 2D position. The projection of this bias will be negligible when the elevation of the aircraft is close to zero, but it will be an important factor when the aircraft is near to the sensor and the elevation is big. This situation is typical in some air traffic control systems, as that of a radar close to an airport with several others far away of that airport. Then the horizontal projection of the measures from the near radar are heavily distorted due to the systematic altitude errors.

The projection in the horizontal plane is implemented using the target elevation, as calculated in Equation (2). This equation is equivalent to search the cross point between the circle defined with radius ρ and the line defining the constant measured altitude. This is shown the Figure 10, where ρ is

the measured range, h and h_m are the ideal and measured altitude respectively. ρ_p is the projected range calculated with the ideal altitude and ρ_{mp} is the projected range calculated with the measured altitude.

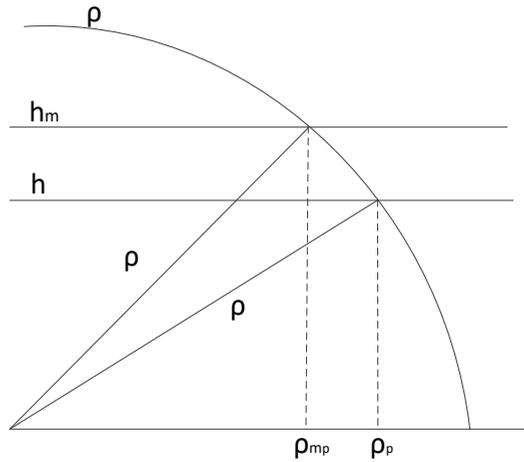


Figure 10. Difference between the projection in the slant range for the same range measure and different altitude.

In [28] a model of the atmosphere for its use in Air Traffic Management (ATM) trajectory prediction, based on ISA model, is defined. The ranges of altitudes of civil aircraft cover the troposphere and the lower stratosphere. The tropopause is the boundary between both layers. Hence there are two different models for different altitudes. In this model, the tropopause is allocated where the barometric altitude is 11,000 m. Below the tropopause the atmospheric model of barometric altitude is a non-linear function that depends on the temperature and the pressure at MSL. Above the tropopause, the model is a linear function because the gradient of temperature is zero.

In the model, the barometric altitude (h_p) is the measured altitude and the geometric altitude (h_g) will be the ideal altitude. The difference between both altitudes will be considered the altitude bias. The model proposed in [28] is:

Below tropopause:

$$h_g = h_p - \Delta H_p + \frac{\Delta T}{\beta_t} \ln \left(\frac{T_0 + \beta_t h_p}{T_0 + \beta_t \Delta H_p} \right) \tag{49}$$

Above tropopause:

$$h_g = h_{g,trop} + \frac{T_0 + \Delta T + \beta_t h_{p,trop}}{T_0 + \beta_t h_{p,trop}} (h_p - h_{p,trop}) \tag{50}$$

where ΔH_p is the barometric altitude offset with respect to ISA (which depends on the difference between current pressure and standard pressure at mean sea level), ΔT is the difference between current temperature and standard temperature at mean sea level, T_0 is the temperature of the standard atmosphere at mean sea level, β_t is the temperature gradient and $h_{g,trop}$ is the geometric altitude where the barometric altitude is $h_{p,trop} = 11,000$ m (which should be calculated with Equation (49) to guarantee geometric height continuity).

To reduce the impact of altitude biases in measurement projection in the horizontal plane ΔT and ΔH_p must be estimated and corrected. Next we will derive a parametric model for these terms. Note the altitude bias effect is important when the elevation of the aircraft is big (high altitude and short

range). We will next fit and show the consistency of a linear model between the geometric altitude and the barometric altitude based on the previously presented model, resulting in:

Below tropopause: we may approximate linearly $\ln(1+x) \approx x$ when $x \approx 0$. Using this approximation in $\ln\left(1 + \frac{T_0 + \beta_t h_p}{T_0 + \beta_t \Delta H_p} - 1\right)$ from (49) we can derive:

$$h_g = h_p - \Delta H_p + \frac{\Delta T}{\beta_t} \left(\frac{T_0 + \beta_t h_p - T_0 + \beta_t \Delta H_p}{T_0 + \beta_t \Delta H_p} \right) \quad (51)$$

After some algebra we have:

$$h_g = (h_p - \Delta H_p) \left(1 + \frac{\Delta T}{T_0 + \beta_t \Delta H_p} \right) \quad (52)$$

and then we can solve for h_p , resulting:

$$h_p = \frac{h_g}{1 + \frac{\Delta T}{T_0 + \beta_t \Delta H_p}} + \Delta H_p \quad (53)$$

Above tropopause: Regrouping the terms in Equation (50), and approximating $h_{g,trop}$ using (52), the following relation is obtained:

$$h_g = h_{g,trop} + \left(1 + \frac{\Delta T}{T_0 + \beta_t h_{p,trop}} \right) (h_p - h_{p,trop}) \quad (54)$$

where:

$$h_{g,trop} = (h_{p,trop} - \Delta H_p) \left(1 + \frac{\Delta T}{T_0 + \beta_t \Delta H_p} \right) \quad (55)$$

Then, we can solve again for h_p , resulting:

$$h_p = \frac{h_g - h_{g,trop}}{1 + \frac{\Delta T}{T_0 + \beta_t h_{p,trop}}} + h_{p,trop} \quad (56)$$

In this section h_g is always the ideal geometric altitude (called h in Equation (3)) and h_p is the barometric altitude (measured by the barometer, called h_m in Equation (3), assuming negligible altitude noise). Changing the notation to that of Equation (3) the resultant model is:

Below tropopause:

$$h_m = \frac{h}{1 + \frac{\Delta T}{T_0 + \beta_t \Delta H_p}} + \Delta H_p \quad (57)$$

Above tropopause:

$$h_m = \frac{h - h_{g,trop}}{1 + \frac{\Delta T}{T_0 + \beta_t h_{p,trop}}} + h_{p,trop} \quad (58)$$

In both cases, the model has an offset height and a height gain. In Figure 11 the bias associated to the more exact EUROCONTROL model [28] in Equations (49) and (50), and the linearized model are represented (in Equations (57) and (58)) vs. the geometric altitude (h). Those two models use the same values for the parameters $\Delta H_p = 300$ m and $\Delta T = 15^\circ$, and the linearization has very low error at low altitudes but at high altitudes the errors between both models can be around 100 m. Meanwhile, the black line (labelled expected estimation) represents a least squares regression of the EUROCONTROL model using the model in Equations (57) and (58) with parameters ΔH_p and ΔT . The optimal values of the estimated values, assuming uniform distribution of heights, are $\Delta H_p = 298.72$ m and $\Delta T = 17.06$ °C. Although those are not the actual values used for the EUROCONTROL model, the simplified model is able to adjust quite finely to it, and therefore seems a

promising model for altitude bias error estimation. It should be noted the purpose of this model is not obtaining a very accurate estimate of ΔH_p and ΔT , but to be able to approximate correctly the altitude bias relation with geometric height.

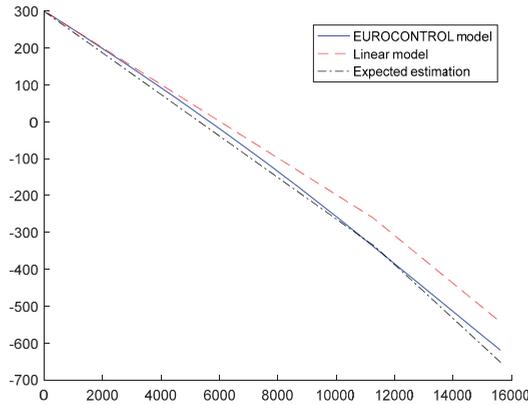


Figure 11. Representation of altitude bias for EUROCONTROL model, linearized model and expected estimation for $\Delta H_p = 300$ m and $\Delta T = 15^\circ$.

6. Bias Estimation with Opportunity Traffic

As the bias error varies slowly with time, the error model parameters must be estimated dynamically. In a radar network the main real time information source are the measures taken from the opportunity traffic and they are used for bias estimation. With the measures of a single radar the biases cannot be determined because there is not a reference in order to get the projection of the biases over the horizontal measurement plane. The projection of the biases depends on the position of the aircraft and the radar.

In this section, we will describe a method for the estimation of the bias parameters of a pair of radars with overlapped coverage. The purpose of the described method is not to describe an optimal/scalable process implementable in a real sensor network with more than two radars, but to analyze in the results section the adequacy of the proposed bias models and their effectiveness to remove systematic errors. More advanced methods may be found in the literature [6–9,13–17], and the application/extension of some of them will be part of our future research.

From the previous sections, bias model parameters of the i -th radar can be arranged in a 12D vector:

$$b_i = [\Delta\theta_0 \ s_{ant} \ t_{axis} \ s_{axis} \ \Delta R_x \ \Delta R_y \ s_{enc,s} \ s_{enc,c} \ \Delta\rho_0 \ \alpha_1 \ \alpha_2 \ \alpha_3]'$$
 (59)

On the other side, altitude bias terms due to the atmosphere conditions are common to every radar and they are included in other bias vector:

$$b_{atm} = [\Delta H_p \ \Delta T]'$$
 (60)

In our simple example method biases will be estimated with the difference of measurements (pseudomeasurements) of the same aircraft taken by different sensors at the same time. This pseudomeasurements will be obtained in the common plane subtracting the horizontal projection of the same aircraft measures. As in real scenarios the measurements of the radars are not synchronized the measurements of one of the sensors need to be interpolated to the time of the other radar, following a process as proposed in [7]. This interpolation is made between two measures from the same radar to the measure time from the other radar (between the times from the other two measures).

Assuming that biases are small the projection of the bias terms may be approximated with a linearized function. With this linearization the measurement model for each of the plots projected in stereographic coordinates can be expressed as:

$$X_{m,i} = \begin{pmatrix} x_{m,i} \\ y_{m,i} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + H_{b_i} b_i + H_{b_{,atm}} b_{atm} \tag{61}$$

where (x,y) is the ideal horizontal position of the aircraft and $(x_{m,i}, y_{m,i})$ is the measured position of the aircraft by the i -th radar while H_{b_i} and $H_{b_{,atm}}$ are the bias projection matrices.

The projection matrices (H_{b_i} and $H_{b_{,atm}}$) are a composition of the linearized change of coordinates from Equation (1), the linearization of the projection to the stereographic plane, and the linearization of the bias models for each of the bias components, so H_{b_i} and $H_{b_{,atm}}$ may be expressed as:

$$H_{b_i} = \begin{bmatrix} \frac{dx}{d\Delta\theta_0} & \frac{dy}{d\Delta\theta_0} \\ \frac{dx}{ds_{ant}} & \frac{dy}{ds_{ant}} \\ \frac{dx}{dt_{axis}} & \frac{dy}{dt_{axis}} \\ \frac{dx}{ds_{axis}} & \frac{dy}{ds_{axis}} \\ \frac{dx}{d\Delta R_x} & \frac{dy}{d\Delta R_x} \\ \frac{dx}{d\Delta R_y} & \frac{dy}{d\Delta R_y} \\ \frac{dx}{ds_{enc,s}} & \frac{dy}{ds_{enc,s}} \\ \frac{dx}{ds_{enc,c}} & \frac{dy}{ds_{enc,c}} \\ \frac{dx}{d\Delta\rho_0} & \frac{dy}{d\Delta\rho_0} \\ \frac{dx}{d\alpha_1} & \frac{dy}{d\Delta\alpha_1} \\ \frac{dx}{d\alpha_2} & \frac{dy}{d\Delta\alpha_2} \\ \frac{dx}{d\alpha_3} & \frac{dy}{d\Delta\alpha_3} \end{bmatrix} \quad H_{b_{i,atm}} = \begin{bmatrix} \frac{dx}{d\Delta H_P} & \frac{dy}{d\Delta H_P} \\ \frac{dx}{d\Delta T} & \frac{dy}{d\Delta T} \end{bmatrix} \tag{62}$$

6.1. Biases Values Initialization

The initial values of the biases will be estimated using a LSE method using a small number of measurement (for example the measurements received in an antenna scan period). As it is proposed in [6] a pseudomeasurement (X_b) of the bias vector will be constructed and the (64) will be solved with the LSE method.

$$X_b = X_{m,1} - \hat{X}_{m,2} \tag{63}$$

$$X_b = H_b b + w \tag{64}$$

where $X_{m,1}$ is a measure from the radar 1, $\hat{X}_{m,2}$ is the interpolation from two measures from the radar 2 to the same time of $X_{m,1}$, w is the measurement white Gaussian noise projected to stereographic plane and:

$$H_{b,i} = [H_{b1}, -H_{b2}, H_{b1,atm} - H_{b2,atm}] \tag{65}$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_{atm} \end{bmatrix} \tag{66}$$

where $H_{b,i}$ is the projection matrix of the i -th measurement. The biases are estimated using the pseudoinverse matrix [5]:

$$b_0 = [H'R^{-1}H]^{-1} H'R^{-1}X_b \tag{67}$$

where:

$$H = \begin{bmatrix} H_{b,1} \\ \vdots \\ H_{b,i} \\ \vdots \\ H_{b,n} \end{bmatrix} R = \text{diag}(R_1 \dots R_n) \tag{68}$$

R_i is the noise covariance matrix of each pseudomeasurement (it is the addition of the noise covariance matrix from $X_{m,1}$ and $\hat{X}_{m,2}$ of the i -th measurement) and n is the number of pseudomeasurements used in the initialization. The initial covariance matrix (P_0) used in the next section is calculated with the projection of the noise over the biases [5].

$$P_0 = (H'R^{-1}H)^{-1} \tag{69}$$

6.2. Recursive Bias Estimation

After initialization, the values of the biases are estimated recursively with an Extended Kalman Filter. Biases are considered constant, so the prediction matrix (Φ) is the identity matrix. Sub-index k indicates the temporal index of the state and the diacritic circumflex indicates that it is a predicted state. The predicted state and its covariance matrix at the time of the current pseudomeasurement are:

$$\begin{aligned} \hat{b}_k &= \Phi b_{k-1} \\ \hat{P}_k &= P_{k-1}\Phi P'_{k-1} + Q_{k-1} \end{aligned} \tag{70}$$

In simulated data where the biases are constant, the noise plant covariance matrix can be set to zero, but in real scenarios the biases will be quasi-constant and the Q matrix must be set with small values of covariance. After that the residual and its covariance matrix must be estimated:

$$\begin{aligned} X_{b,k} &= X_{m,1} - X_{m,2} \\ r_k &= X_{b,k} - H_{b,k}\hat{b}_k \\ S_k &= H_b\hat{P}_kH'_b + R_k \end{aligned} \tag{71}$$

With these previous values, the filter gain is calculated and then the updated state so the updated covariance of the filter are calculated:

$$\begin{aligned} K_k &= \hat{P}_kH'_{b,k}S_k^{-1} \\ b_k &= \hat{b}_k + K_k r_k \\ P_k &= (I - K_kH_{b,k})\hat{P}_k \end{aligned} \tag{72}$$

6.3. Bias Observability Discussion

Ideally the observability of the bias parameters in the scenario can be tested with the matrix $[H'R^{-1}H]$ where H and R is composed by all the $H_{b,k}$ and R_k as in Equation (68). The bias parameters will be observable if this matrix is a positive definite matrix [15]. In the initialization phase this matrix is composed in order to calculate Equation (67). In the recursive bias estimation phase this matrix is never calculated but the observability can be tested composing this matrix with all the measurements. The dimensions of the resultant matrix becomes in an unaffordable computational load. The objective of this discussion is show the geometrical requirements (presented in the next paragraphs) that a radar scenario should have for the stability of the estimation algorithms. For a more rigorous mathematical analysis of the observability in radar registration the modified Fischer Information Matrix can be studied [29].

In the best case the targets have diversity in range, azimuth and altitude and the biases are perfectly determined. In many real cases the opportunity traffic is not uniformly distributed in the airspace as most of the aircraft are in the upper flight levels. LSE method tries to minimize the global errors for all the tracks. If most of the tracks are concentrated in the same zone, the method will estimate the bias model parameters to get the better approximation of the corrections for these tracks, but the estimated parameters could lead to a bad bias correction for the aircraft in areas with lower density traffic.

Another issue about the bias observability is the variability of the measurement values. If the sensors are widely separated in space, tracks with common coverage are concentrated in a small range of azimuth values. For example, in the azimuth bias model presented in Section 3, we may observe that the effect of the antenna squint and the axis squint are similar. The single difference between them is that axis squint depends on the azimuth of the target. If every target has similar azimuth, the effects of both bias terms are difficult to separate and their estimation is highly correlated. In this case, the position measures biases are well corrected for the tracks used in the estimation (inside common sensor coverage). But for other tracks with azimuth far from common coverage, there could be position errors because the azimuth bias dependence is not well estimated.

7. Results

In this section we will include results to show the performance improvement due to the use of our improved bias models both using simulated data (in Section 7.1) and real data (in Section 7.2). Also, the consistency of the estimator using this model will be assessed.

The evaluation of the results will be made evaluating the Root Mean Square (RMS) value of the difference between the positions of the measurements from different radars at the same time for the same target:

$$error_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left((x_{i,1} - x_{i,2})^2 + (y_{i,1} - y_{i,2})^2 \right)} \quad (73)$$

where $(x_{i,1}, y_{i,1})$ and $(x_{i,2}, y_{i,2})$ are the i -th measures from first and second radar for the same target at the same time. This method is used because the evaluation of this model done principally with real data.

Another way to test the results for real data is to have access to ADS-B or WAM data that can be supposed unbiased compared with radar measurements. As the only available information is the measurements of the aircraft, only the errors of the corrected plots can be evaluated, but it is almost impossible know the real value of the biases. In route tracking in ATC it is more important the alignment between plot than the accuracy in position of the measurement.

For simulated data the value of the biases is known and the results are presented in convergence graphics (as a function of the number of samples used in the simulation).

7.1. Simulated Scenario Results

In order to test the observability of the different biases modelled in previous sections, the estimation method will be tested with a simulated scenario. In this scenario there are two radars in position $(0, 0)$ and $(50, 0)$ (in NM). The scenario has 1000 point uniformly distributed in a square area with a side of 400 NM centered at $(0, 0)$. The maximum altitude of these points is 15,000 m.

In this case the simulated measurements are synchronized and the noise of the measurements are white Gaussian noise with zero mean with standard deviation of 75 m in range and 0.05° in azimuth. This test is made using Cartesian coordinates with the flat Earth model.

The measures obtained by the radars are biased using the previously developed models with all the parameters. The evaluation of the results is made using the RMS error in position of all the corrected measurement in Equation (73). As the ideal position of the points is known, the error will be calculated between the corrected measurement and the ideal position.

In this simulated scenario, the RMS error of the measured plots is 434.72 m and the RMS error of the corrected plots is 126.13 m. As in the simulated data the ideal position of the aircraft is known, we can measure the errors produced only by the noise. In this case, the RMS error produced just by the noise is 120.08 m. As the RMS error of the corrected plots is almost the RMS error produced by the noise of the measurements, it is proved the observability of the model when the plots are well distributed. With a bigger number of samples, the RMS error of the estimation is nearer to the RMS error produced by the noise.

In the previous simulated scenario the bias parameter used in the generation of the simulated position are shown in Table 1. The next paragraphs will show the estimation of some parameters using different models but always generating the simulated plots with the complete model.

Table 1. Values of the biases used in the simulation for the radar 1.

Parameters	RMS Error
$\Delta\rho_0$	100 m
α_1	10^{-3} m/m
α_2	10^{-9} m/m ²
α_3	1.15
θ_0	0.04°
s_{ant}	0.5°
β_{axis}	0.4°
α_{axis}	45°
β_{enc}	0.5°
α_{enc}	90°
$\Delta R/R$	5×10^{-4}
α_{exc}	45°
ΔH_p	-500 m
ΔT	15 °C

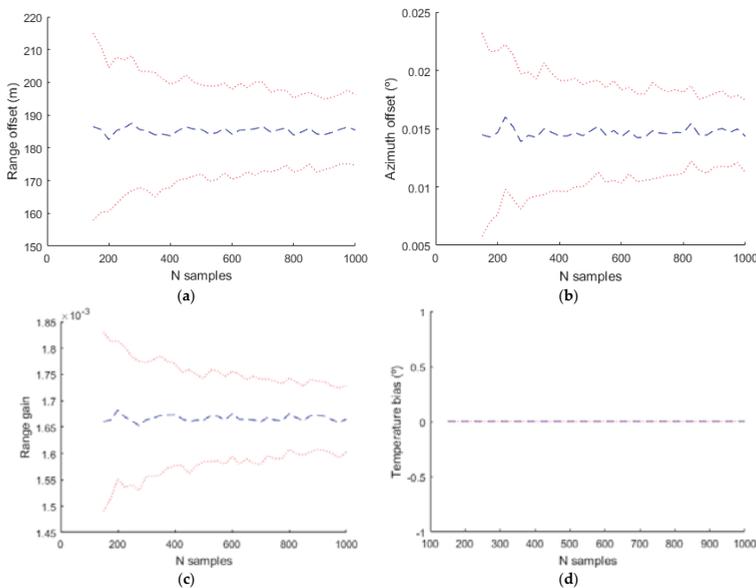


Figure 12. Estimated bias values for Radar 1 using only the basic model ($\Delta\rho_0$, α_1 and θ_0). (a) Range offset; (b) Azimuth offset; (c) Range gain; (d) Temperature bias.

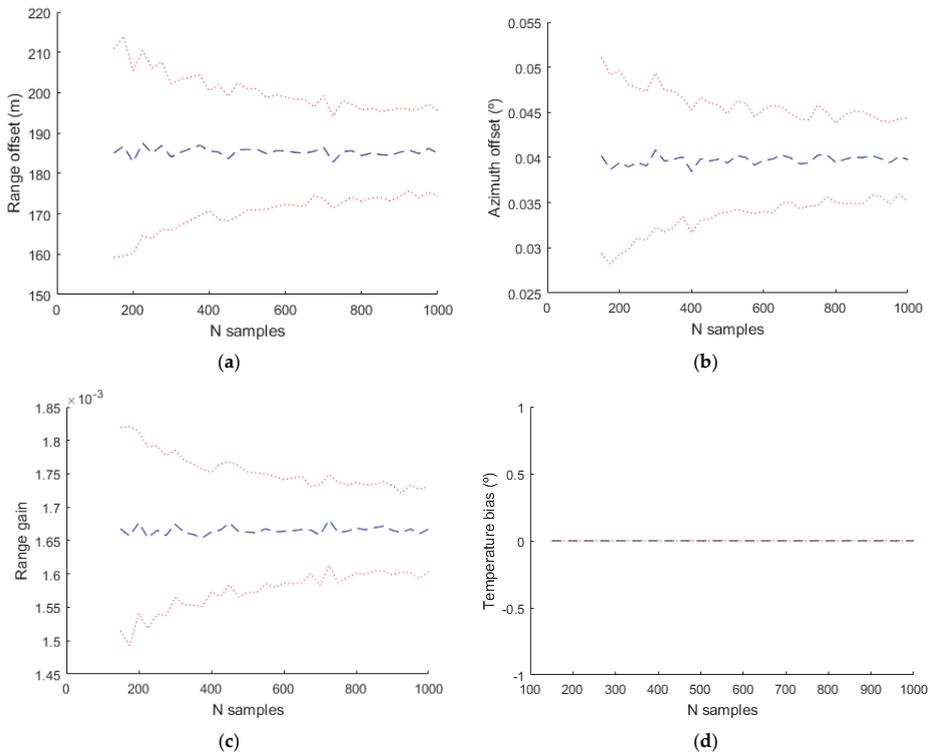


Figure 13. Estimated bias values for Radar 1 using only the model with the parameters $\Delta\rho_0$, α_1 , θ_0 , s_{ant} , β_{axis} and α_{axis} . (a) Range offset; (b) Azimuth offset; (c) Range gain; (d) Temperature bias.

In the evaluation of the simulated data, the complete model with the values indicated in Table 1 has been used in the generation of the plots, but different bias models with more or less parameters have been used in the estimation. The next figures shows the mean estimated value of the biases with a blue line and the mean plus/minus the standard deviation of the estimation with a dotted red line.

The basic model used classically in the bias estimation literature uses only $\Delta\rho_0$, α_1 and θ_0 . The obtained results are shown in Figure 12. The temperature bias isn't estimated with this model and in the figure corresponding to this bias the value and the deviation is zero. With a more complex model where the parameters used in the estimation are $\Delta\rho_0$, α_1 , θ_0 , s_{ant} , β_{axis} and α_{axis} the estimated values are shown in Figure 13.

Estimating different parameters than in the previous simulation ($\Delta\rho_0$, α_1 , θ_0 , β_{enc} , α_{enc} , $\Delta R/R$ and α_{exc}) the results are slightly different as it can be seen in Figure 14. In these three simulations it can be seen that the range and azimuth estimations are different from the real values. This is due to the fact that there are parameters that affect range and azimuth that aren't included in the estimation model and the included parameters values compensate the effect of the not included parameters.

Finally Figure 15 shows the estimated values for the complete model used in the generation. In this case all the parameters are estimated and the result are near to the real values (shown with a black dotted line).

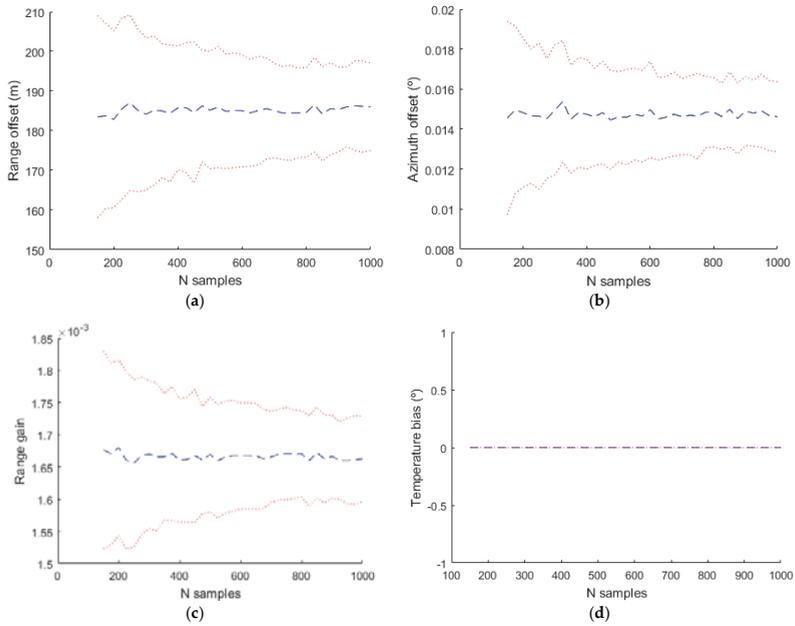


Figure 14. Estimated bias values for Radar 1 using only the model with the parameters $\Delta\rho_0$, α_1 , θ_0 , β_{enc} , α_{enc} $\Delta R/R$ and α_{exc} . (a) Range offset; (b) Azimuth offset; (c) Range gain; (d) Temperature bias.

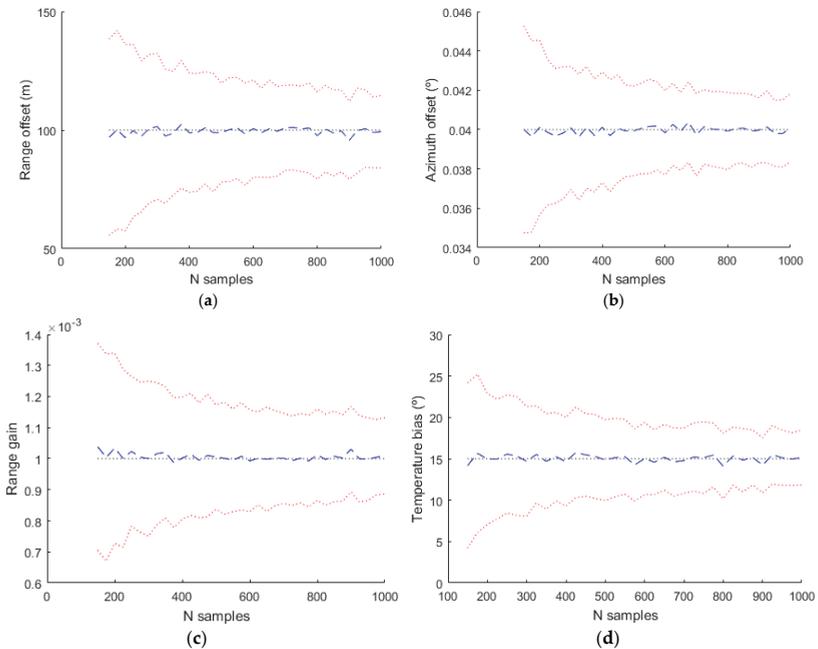


Figure 15. Estimated bias values for Radar 1 using the complete model. (a) Range offset; (b) Azimuth offset; (c) Range gain; (d) Temperature bias.

7.2. Real Data Results

The models have been tested with data from two real radars. The maximum range of both radars is bigger than 150 NM and the separation between both radars is 52 NM. This configuration gives enough common coverage and avoids observability problems due to the sensor separation. Radar measurements have been transformed to local Cartesian coordinates and projected to a common stereographic plane. The center of the stereographic projection (point (0, 0) in Figure 12) is far away from the radars in order to evaluate if the models work with the rotation, translation and scaling produced by the stereographic projection.

The real scenario has 227 aircraft distributed in the airspace in different flight phases. As it can be seen in Figure 16 tracks are not uniformly distributed in the airspace. The major part of the measures is concentrated in the bottom-left corner of the common radar coverage. As the algorithm minimizes the mean square error the measurement of this corner will be better aligned than the measurements from other parts of the scenario, as described in Section 6.3.

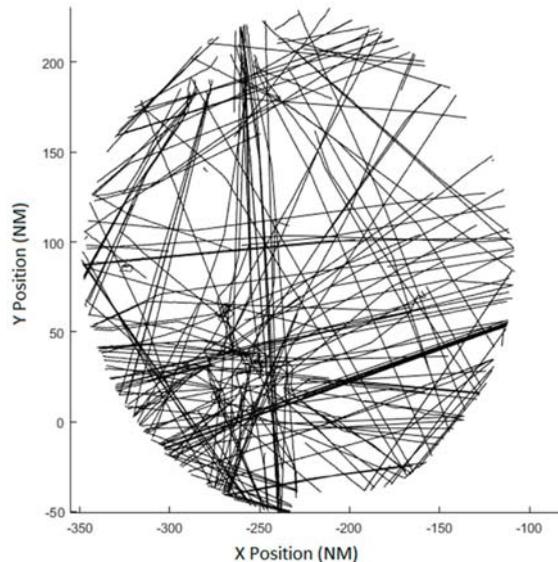


Figure 16. Real scenario with 227 tracks measured by two radars.

As the biases values are not known, the performance of the model will be evaluated estimating the RMS value of the deviation in measured position of the same track at the same time from two different radars (73). The evaluation will compare the deviation of the positions for all the measures in the common coverage. The RMS error of the measurement noise of the radar is 160.95 m for the first radar and 88.92 m for the second radar. The RMS value of the difference of uncorrected measured positions is 532.45 m.

7.3. Basic Model

In [6,7] a simplified bias model is used for range and azimuth without bias in altitude. The biases used in this estimation are the range offset ($\Delta\rho_0$), the range gain (α_1) and the azimuth offset ($\Delta\theta_0$). With these three parameters the majority of the error produced by the biases is corrected. With these models, the deviation of the bias errors σ_b is reduced up to 286.60 m obtained with Equation (73). In Figure 17a,b two illustrative trajectories are represented. The asterisks mark the measured position

and the circles mark the corrected position. Both trajectories are allocated in the bottom left corner of the scenario and the raw plots of radar 2 are displaced in azimuth in counter-clockwise direction. The algorithm corrects the plots and displaces the plots towards the correct azimuth. In Figure 17c several trajectories placed in top-right corner (low density traffic area) are represented. Raw plots are aligned and the corrected plots are displaced clockwise. In this scenario is easy to see that the simple model is not good enough to estimate and align the plots of the radars for the whole radar coverage. In this scenario, the trajectories of the Figure 17a,b are separated almost 180° in azimuth from trajectories of the Figure 17c. The azimuth dependence of the biases is clearly shown in these figures. This dependence is included in the complete azimuth bias model of Equation (34).

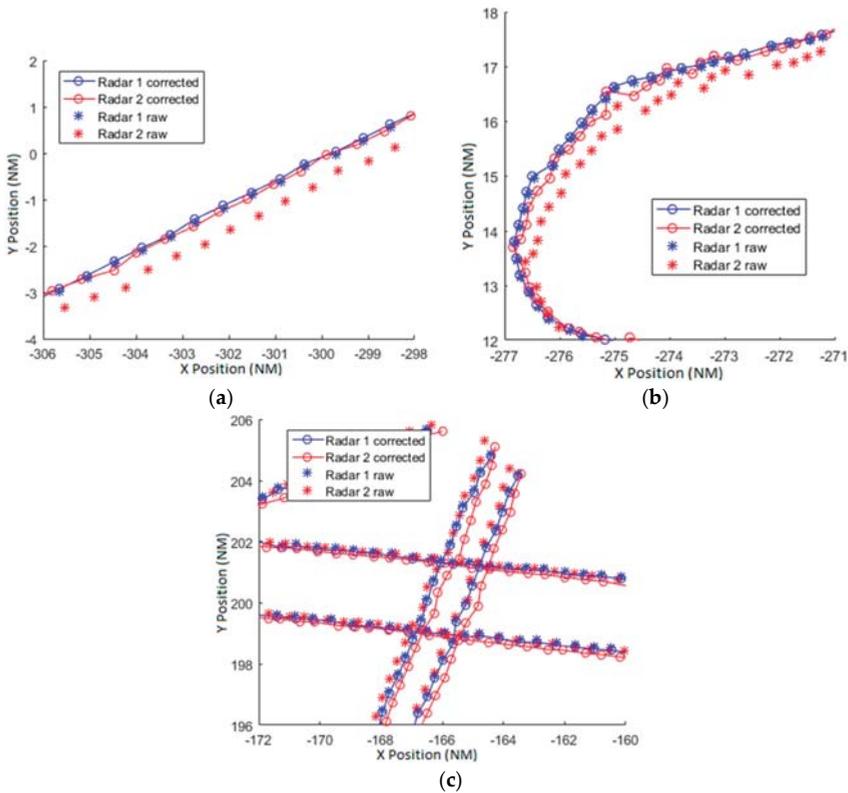


Figure 17. (a) Straight trajectory; (b) Take off trajectory corrected with basic model; (c) Several trajectories allocated at the upper-right corner of the scenario corrected with the basic model.

7.4. Complete Model

Next, we will show the results with variations of the complete model developed in this paper. Several configurations of the model have been used with the objective to test the final correction contribution of each parameter of the model. In Table 2 the error deviation due to the biases is expressed for different configurations of the models (in the first column the bias parameters used in each configuration are indicated). The first row is the basic model from the previous section. Second row estimates the biases due to the deviation of the antenna and the rotation axis. Third row estimates the biases produced by the encoder. Finally the fourth row estimates all the biases presented in the model.

Table 2. Bias error deviation for different configuration of the model parameters.

Parameters	RMS Error
Without correction	532.45 m
$[\Delta\rho, \alpha_1, \Delta\theta]$ (basic model)	286.60 m
$[\Delta\rho, \alpha_1, \Delta\theta, s_{ant}, \beta_{axis}, \alpha_{axis}]$	260.65 m
$[\Delta\rho, \alpha_1, \Delta\theta, s_{enc}, \alpha_{enc}, \Delta R/R, \alpha_{ecc}]$	248.22 m
$[\Delta\rho, \alpha_1, \alpha_2, \alpha_3, s_{ant}, s_{axis}, t_{axis}, \Delta\theta, s_{enc}, \alpha_{enc}, \Delta R/R, \alpha_{ecc}, \Delta H_p, \Delta T]$	242.19 m

The encoder eccentricity seems to be a very important parameter in the bias estimation. For example if the encoder has a diameter of 25 cm and an error in the rotation center of 0.25 mm produces a maximum bias of 200 m at a range of 200 km. As could be expected, the estimation with all the parameters is the best one and in the Figure 18 (equivalent to Figure 17) the corrected tracks are shown. In this case, as the complete model has azimuth dependent biases, every track in every azimuth is better corrected in comparison with the simple model. But even reduced parameters bias models result in relevant accuracy gains with respect to the usual basic model.

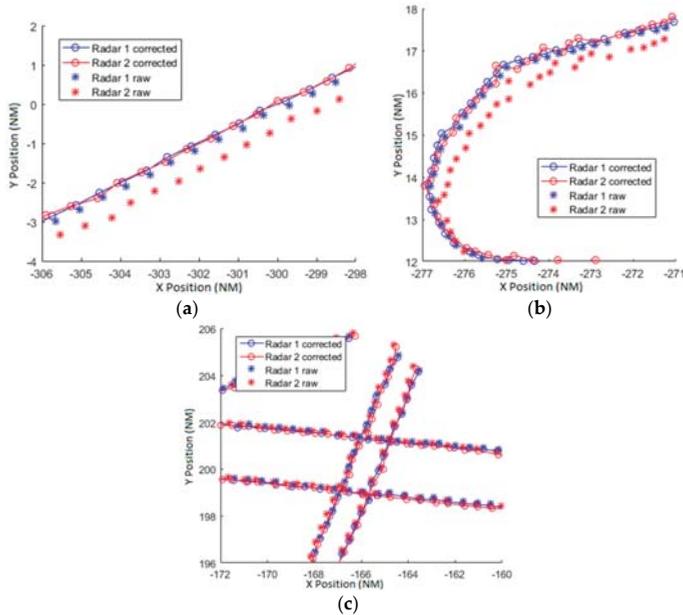


Figure 18. (a) Straight trajectory (b) Take off trajectory corrected with basic model (c) Several trajectories allocated at the upper-right corner of the scenario corrected with the complete model.

The algorithm determines bias parameters for both radars to minimize the mean squared error between the ideal position and the corrected position. As consequence of results we should conclude that (locally in the in the shown areas) the radar 1 is better aligned to ideal position than radar 2, although measures from both radars are corrected.

8. Conclusions

A complete model for radar biases in ATC has been developed in the paper considering both mechanical errors in the radar installation and errors in the physical propagation and atmospheric parameters. Typically only a basic bias model is considered in bias estimation and correction. As has

been demonstrated with our results, with these simplified model parameters the major part of the errors can be corrected. In real scenarios the errors of the measures cannot be approximated only with the basic bias model. In online applications where the execution time is an important feature, the simple model reduces the major part of the errors. In offline applications (such as [30]) where the accuracy of the estimations is more relevant than the execution time, the number of estimated parameters can be increased reducing the final bias error. As the number of parameters is increased, the degrees of freedom to interpolate generic bias patterns is increased too.

The bias estimation methods in general minimize the MSE and are better adjusted where the traffic density is bigger. With the basic model the biases are only adjusted to these higher density areas, leaving lower traffic density with badly corrected measures. With the complete biases model there are more degrees of freedom and the estimations will be better adjusted for areas where there is lower density.

Analyzing the results using different bias configurations (estimation with different bias parameters) it can be observed that the basic biases will correct the major part of the systematic errors. The encoder eccentricity has a significant effect on the azimuth that does not depend on the target elevation and the use of this parameter will align the plots better than with the simple model. Also, trajectories near to one of the radars are highly affected by altitude errors.

For future work two topics derived from this paper can be studied. On the one hand the distribution of the plots seems to be an important topic. An estimation method will be studied in order to try to make the density of the plots in a given area not so critical for the precision of the estimations. On the other hand, the parametrized model developed in this paper assumes that the atmospheric conditions are constant in all the airspace but the temperature and pressure at MSL change slowly with position. In the future we will also study the extension of the model with slow changing curves for pressure and temperature, and the potential associated improvements. Also, the estimation of transponder error, maybe exploiting ideas similar to the ones in [8] will be addressed.

Acknowledgments: The research has been partially funded by UPM's project 'Tecnologías Avanzadas para la Monitorización y Gestión Remota del Tráfico Aéreo de Vehículos Pilotados y no Pilotados' (RP1509550C02); by FUNDETEL project 'Tecnologías y Sistemas de Control y Gestión de Tráfico Aéreo' (GPD.SJC.001), and by the Spanish Ministry of Economy and Competitiveness through grant TEC2014- 57022-C2-1-R.

Author Contributions: Ángel J. Jarama and Jaime López-Araquistain performed the main part of the work: definition of the bias model, design of estimation algorithms and proofs. Ángel J. Jarama concentrated principally on theoretical aspects and wrote the paper while Jaime López-Araquistain focused on experimental aspects. Gonzalo de Miguel and Juan A. Besada concentrated in the general design of the bias correction process, the methodology to demonstrate its validity and the supervision of results.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Neven, W.H.L.; Quilter, T.J.; Weedon, R.; Hogendoorn, R.A. *Wide Area Multilateration*; Report on EATMP TRS 131/04, Version 1.0; National Aerospace Laboratory NLR: Noordoostpolder, The Netherlands, 2004.
2. Eurocontrol. *Surveillance Data Exchange, Part 12: Category 021*; ADS-B Reports; SURET1.ST05.2000-STD-12-01; European Organisation for the Safety of Air Navigation: Brussels, Belgium, 2009.
3. Stevens, M.C. *Secondary Surveillance Radar*; Artech House: Montreal, QC, Canada, 1988.
4. Blackman, S.; Popoli, R. *Design and Analysis of Modern Tracking Systems*; Artech House: Norwood, MA, USA, 1999.
5. Kay, S.M. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*; Prentice Hall: Bergen County, NJ, USA, 1993.
6. Lin, X.; Bar-Shalom, Y.; Kirubarajan, T. Exact multisensor dynamic bias estimation with local tracks. *IEEE Trans. Aerosp. Electron. Syst.* **2004**, *40*, 576–590.
7. Lin, X.; Bar-Shalom, Y.; Kirubarajan, T. Multisensor multitarget bias estimation for general asynchronous sensors. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 899–921.

8. Besada, J.A.; De Miguel, G.; Bernardos, A.M.; Casar, J.R. Generic multisensor multitarget bias estimation architecture. *IET Radar Sonar Navig.* **2012**, *6*, 365–378. [CrossRef]
9. Pu, W.; Liu, Y.F.; Yan, J.; Zhou, S.; Liu, H.; Luo, Z.Q. A two-stage optimization approach to the asynchronous multi-sensor registration problem. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 3271–3275.
10. Zhou, Y.; Leung, H.; Yip, P.C. An exact maximum likelihood registration algorithm for data fusion. *IEEE Trans. Signal Process.* **1997**, *45*, 1560–1573. [CrossRef]
11. Ignagni, M. An alternate derivation and extension of Friendland’s two-stage Kalman estimator. *IEEE Trans. Autom. Control* **1981**, *26*, 746–750. [CrossRef]
12. Guiar, C.N.; Lansing, F.L.; Riggs, R. *Antenna Pointing Systematic Error Model Derivations*; TDA Progress Report Jet Propulsion Laboratory: Pasadena, CA, USA, 1987; pp. 36–46.
13. Belfadel, D.; Osborne, I.; Richard, W.; Bar-Shalom, Y. Bias Estimation and Observability for Optical Sensor Measurements with Targets of Opportunity. *J. Adv. Inf. Fusion* **2014**, *9*, 59–74.
14. Chen, L.; Wang, G.H.; Progni, I.F. Unified registration model for both stationary and mobile 3D radar alignment. *J. Electr. Comp. Eng.* **2014**, *2014*, 29. [CrossRef]
15. Bar-Shalom, Y.; Willett, P.K.; Tian, X. *Tracking and Data Fusion*; YBS Publishing: Storrs, CT, USA, 2011.
16. Helmick, R.E.; Rice, T.R. Removal of alignment errors in an integrated system of two 3-D sensors. *IEEE Trans. Aerosp. Electron. Syst.* **1993**, *29*, 1333–1343. [CrossRef]
17. Nageswara Rao, K.; Soma, P. Estimation of Systematic Errors in Angles of Tracking Antennae. In Proceedings of the Fourth International Symposium, Space Mission Operations and Ground Data Systems-SpaceOps, Munich, Germany, 16–20 September 1996; p. 877.
18. Torge, W.; Müller, J. *Geodesy*; Walter de Gruyter: Berlin, Germany, 2012.
19. Hudgens, J.M.; Cawthon, G.M.; Suwane, G.A. Gimbals for Antenna & Radome Measurement: Demanding Applications Drive Innovative Architecture, Remarkably Higher Accuracy, AMTA 2014. In Proceedings of the 36th Annual Symposium of the Antenna Measurement Techniques Association, Tucson, AZ, USA, 12–17 October 2014.
20. Qin, S.; Huang, Z.; Wang, X. Optical angular encoder installation error measurement and calibration by ring laser gyroscope. *IEEE Trans. Instrum. Meas.* **2010**, *59*, 506–511. [CrossRef]
21. Zhao, R.; Zhang, Z.; Tie, J. Influence of Encoder Eccentricity on Speed Measurement and Elimination Approach. In Proceedings of the IEEE International Conference on Network Computing and Information Security (NCIS), Guilin, China, 14–15 May 2011; pp. 63–66.
22. Fischer, W.L.; Muehe, C.E.; Cameron, A.G. *Registration Errors in a Netted Air Surveillance System*; Technical Report for Massachusetts Institute of Technology Lincoln Lab: Lexington, MA, USA, 1980.
23. Bean, B.R. The radio refractive index of air. *Proc. IRE* **1962**, *50*, 260–273. [CrossRef]
24. Barton, D.K. *Radar System Analysis and Modelling*; Artech House: Norwood, MA, USA, 2004.
25. Zeng, Y.; Blahak, U.; Neuper, M.; Jerger, D. Radar beam tracing methods based on atmospheric refractive index. *J. Atmos. Oceanic Technol.* **2014**, *31*, 2650–2670. [CrossRef]
26. Abbud, J.; De Miguel, G. Joint Target Tracking and Systematic Error Correction for Wide Area Multilateration. In *Electronic Navigation Research Institute Air Traffic Management and Systems*; Lecture Notes in Electrical Engineering Springer: Tokyo, Japan, 2014; Volume 290, pp. 175–191.
27. Abbud, J.M.; De Miguel, G.; Besada, J. Correction of systematic errors in Wide Area Multilateration. In Proceedings of the 2011 Tyrrhenian International Workshop on Digital Communications-Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV), Capri, Italy, 12–14 September 2011; pp. 173–178.
28. Poles, D. *Revision of Atmos Navigation Here Model in BADA Aircraft Performance Model*; EEC Technical/Scientific Report No. 2010-001; EUROCONTROL Experimental Centre: Paris, France, 2010; pp. 9–17.
29. Fortunati, S.; Gini, F.; Greco, M.S.; Farina, A.; Graziano, A.; Giompapa, S. On the identifiability problem in the presence of random nuisance parameters. *Signal Proc.* **2012**, *92*, 2545–2551. [CrossRef]
30. Besada, J.; Soto, A.; de Miguel, G.; García, J.; Voet, E. ATC trajectory reconstruction for automated evaluation of sensor and tracker performance. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *28*, 4–17. [CrossRef]



Article

Tracking of Maneuvering Complex Extended Object with Coupled Motion Kinematics and Extension Dynamics Using Range Extent Measurements

Lifan Sun ^{1,2}, Baofeng Ji ^{1,3}, Jian Lan ^{4,*}, Zishu He ² and Jiexin Pu ¹

¹ School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China; Lifan.sun@gmail.com (L.S.); fengbaoji@126.com (B.J.); pujiexin@126.com (J.P.)

² School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; zshe@uestc.edu.cn

³ Institute of Atmospheric Physics, Chinese Academy of Science, Beijing 100029, China

⁴ Center for Information Engineering Science Research (CIESR), School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China

* Correspondence: lanjian@mail.xjtu.edu.cn; Tel.: +86-29-8266-3421

Received: 1 August 2017; Accepted: 19 September 2017; Published: 22 September 2017

Abstract: The key to successful maneuvering complex extended object tracking (MCEOT) using range extent measurements provided by high resolution sensors lies in accurate and effective modeling of both the extension dynamics and the centroid kinematics. During object maneuvers, the extension dynamics of an object with a complex shape is highly coupled with the centroid kinematics. However, this difficult but important problem is rarely considered and solved explicitly. In view of this, this paper proposes a general approach to modeling a maneuvering complex extended object based on Minkowski sum, so that the coupled turn maneuvers in both the centroid states and extensions can be described accurately. The new model has a concise and unified form, in which the complex extension dynamics can be simply and jointly characterized by multiple simple sub-objects' extension dynamics based on Minkowski sum. The proposed maneuvering model fits range extent measurements very well due to its favorable properties. Based on this model, an MCEOT algorithm dealing with motion and extension maneuvers is also derived. Two different cases of the turn maneuvers with known/unknown turn rates are specifically considered. The proposed algorithm which jointly estimates the kinematic state and the object extension can also be easily implemented. Simulation results demonstrate the effectiveness of the proposed modeling and tracking approaches.

Keywords: maneuvering complex extended object; coupled motion kinematics and extension dynamics; Minkowski sum; range extent measurements

1. Introduction

In traditional radar- and sonar-based tracking applications, most target tracking approaches usually made the assumption that the received measurement originated from a point source at each time, i.e., a target is often regarded as a point source. Maneuvering target tracking has been extensively studied and well developed in many articles due to its military and civil applications, which has attracted wide attention [1]. However, with the increased resolution capability of modern sensors, an object should be regarded as extended if one target occupies more than one resolution cell or its extent is not negligible compared with the sensor resolution [2,3]. Specifically, some high resolution radars can resolve individual features on an extended object and provide its kinematic measurements. In addition, target detection, classification and tracking require more and more knowledge of the object extension information in practical applications. Thus, treating it as a point

mass is no longer reasonable, which results in a significant loss of information. Traditional point tracking algorithms are not suitable for many current practical tracking scenarios. In this context, estimating the kinematic state (i.e., position, velocity, acceleration, etc.) and extension (i.e., size, shape and orientation) of an extended object jointly is referred to as extended object tracking [4]. It has drawn wide attention over the past decades and has been widely applied into the tracking of ground vehicles, close airplanes and ships. Specifically for extended object tracking using (partially) unresolvable measurements of the object's scattering centers, many models and approaches have been proposed, including the multiple hypothesis tracking (MHT) [5], the spatial probability distribution model [6], and the probability hypothesis density (PHD) filters [7–10]. A new approach of a random matrix was presented in [11] and improved in [12]. The random-matrix-based approach was also developed to sufficiently characterize time variation and observation distortion [13,14]. In addition, a so-called random hypersurface modeling approach was proposed in [15] to represent the object extensions, as ellipses and star-convex shapes.

Recently, modern and more accurate sensors can also provide the target's range extent (e.g., down-range and cross-range extent) measurements on a single extended object [16]. Using range extent measurements benefits track retention in practical applications [17]. In [18], recognition of convoys is accomplished by estimating the cross-range extension of the object using target range measurements (high range resolution profiling). For extended object tracking using these types of measurements, several modeling and estimation approaches have also been proposed [19–21]. In [22], a new approach based on support functions was proposed to model extended objects. It subsumes the above approaches using range extent measurements and needs no assumption that the orientation of the object is parallel to its velocity vector. However, the support-function-based approach of [22], its variant in [23,24] and other existing approaches have no explicit consideration of extended object's maneuverability (e.g., turn motions).

This paper aims at solving the maneuvering complex extended object tracking (MCEOT) problem using range extent measurements. Compared with traditional maneuvering target tracking, MCEOT emerges as an important and more difficult problem, i.e., it faces two interrelated challenges in practice: kinematic state and object extension uncertainties. However, the extension dynamics (change in size, shape, orientation, e.g., rotation) of a complex object is not necessarily feasible or convenient to be described and modeled, not to mention it is tightly coupled with the centroid kinematics. In view of the above, how to deal with the MCEOT of both centroid kinematics and the extension dynamics using a target's range extent is rarely accounted for in existing literature. Actually, different maneuvers of extended objects are always reflected simultaneously in both centroid kinematics and the extension dynamics, e.g., when an extended object maneuvers (e.g., it starts or ends a maneuver), the object motion and extension will undergo an abrupt change.

To sum up, there is a pressing need for solving the MCEOT problem using range extent measurements. Thus, this paper first proposes a motion and extension dynamic model describing constant-turn maneuvers based on Minkowski sum. It can not only describe the complex extension dynamics of a maneuvering complex extended object (MCEO) accurately and effectively, but also fully considers the close coupling between the centroid kinematics and the extension dynamics. Furthermore, this largely facilitates the derivation and design of an MCEOT algorithm for estimating the centroid kinematic state and object extension jointly. Specifically, the proposed algorithm can be easily implemented to deal with different cases of the turn maneuvers with known/unknown turn rates for MCEOT. Compared with existing approaches, our modeling and tracking approach has the following innovative aspects:

- (a) MCEOT using measurements of target's range extent is first considered explicitly, our approach characterizes not only the evolution of the kinematic state over time, but also the object extension dynamics. More importantly, the coupling between the centroid kinematics and extension evolution (e.g., the close relationship between the turn maneuver of the centroid and the abrupt change of extension) is also explicitly involved.

- (b) The new model has a concise and unified form and it can accurately describe an MCEO with a turn maneuver in both the extension dynamics and the centroid kinematics, i.e., the maneuver model of a complex extended object can be obtained and directly represented by that of several simple sub-objects (decomposed using the Minkowski sum) jointly. In particular, the elliptical maneuvering object model is obtained in this paper, which is a by-product of the proposed approach.
- (c) Based on the Minkowski sum, different parameterizations can be adopted in our unified modeling framework if they are efficient to describe sub-objects' extension dynamics. This does not affect the generality of the proposed approaches for solving MCEOT.
- (d) Due to the concise linear form, the proposed modeling is easily incorporated into a general tracking architecture, in which the exchange of information between centroid kinematics and extension dynamics are sufficiently utilized. This largely facilitates the derivation and design of an MCEOT algorithm for achieving much better estimation performance.

This paper is organized as follows. Section 1 briefly reviews and analyzes the problem existing in the extended object modeling and estimation approach. Section 2 formulates the problem of MCEOT using range extent measurements provided by high resolution sensors. Section 3 proposes a general approach to modeling an MCEO based on Minkowski sum. Different turn maneuvers of a complex extended object (with different turn rates) can be described sufficiently, which has a unified form and is simple to implement. In addition, we propose an MCEO algorithm using the Minkowski-sum-based model for estimating both the kinematic state and extension of an maneuvering extended object, in which different cases of turn maneuvers with known/unknown turn rates for MCEOT are involved and solved explicitly. In Section 4, simulation results are presented to demonstrate the effectiveness of what we propose. The last section concludes the paper.

2. Problem Formulation

For a MCEO, different maneuvers are reflected in both centroid kinematics and extension dynamics jointly. However, the complex extension dynamics are usually difficult, infeasible or inconvenient to describe, not to mention it is tightly coupled with the centroid kinematics when the object maneuvers. Thus, this is rarely considered in the existing literature for object tracking using range extent measurements. In this paper, the aim is to estimate the joint state $x_k = [(x_k^m)^T, (x_k^e)^T]^T$ of an MCEO using range extent measurements. x_k comprises the centroid state x_k^m ($x_k^m = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$, where (x, y) and (\dot{x}, \dot{y}) are the position and velocity in the two-dimensional (2D) Cartesian coordinate system, respectively) and the vector x_k^e describing the complex object extension. Consider the following system model:

$$\begin{aligned} x_k &= Fx_{k-1} + Gw_{k-1}, \\ z_k &= H(x_k, v_k) \end{aligned} \quad (1)$$

where w_{k-1} and v_k denote the process noise and measurement noise, respectively. The MCEO dynamics of Equation (1) describes the centroid state transition and the change of object extension jointly over time, where k is time index. The joint state transition matrix F is composed of both the centroid kinematic state transition matrix F^m and object extension transition matrix F^e . The uncertainty of the MCEO state x_k is embodied in the process noise $w_{k-1} = [(w_{k-1}^m)^T, (w_{k-1}^e)^T]^T$. Thus, the dynamic equation of Equation (1) has actual the following form:

$$\begin{bmatrix} x_k^m \\ x_k^e \end{bmatrix} = \begin{bmatrix} F^m & 0 \\ 0 & F^e \end{bmatrix} \begin{bmatrix} x_{k-1}^m \\ x_{k-1}^e \end{bmatrix} + \begin{bmatrix} G^m & 0 \\ 0 & G^e \end{bmatrix} w_{k-1}. \quad (2)$$

This paper focuses on the case that an MCEO of interest, assumed to be a rigid body, moves with turn maneuvers in 2D Cartesian coordinate system. As a typical target maneuver, the coordinated turn (CT) motion is also referred to the constant turn (CT), which usually has an approximate constant

speed with a constant turn rate [25]. The MCEO dynamics in Equation (2) describes the turn maneuver. Note that different turn maneuvers correspond to different turn rates. In addition, Equation (2) explicitly relates the object extension orientation change to the centroid turn maneuver, which are tightly coupled. The joint transition matrix of the centroid kinematic state and the object extension state is $F = \text{diag}(F^m, F^e)$, where

$$F^m = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{1-\cos(\omega T)}{\omega} \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) \\ 0 & \frac{1-\cos(\omega T)}{\omega} & 1 & \frac{\sin(\omega T)}{\omega} \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) \end{bmatrix}, \quad (3)$$

and ω is the turn rate of the centroid. F^m of Equation (3) is a centroid kinematic state transition matrix for modeling of a maneuvering point target with CT motions, which has been extensively researched and applied. According to Equation (3), the centroid state transition of $\mathbf{x}_k^m = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$ is

$$\mathbf{x}_k^m = F^m \mathbf{x}_{k-1}^m + G^m \mathbf{w}_{k-1}^m, \quad (4)$$

where $G^m = \text{diag}([\frac{T^2}{2}, T]^T, [\frac{T^2}{2}, T]^T)$, and T is the sampling time. Unlike traditional point target tracking, the key to successful MCEOT using range extent measurements lies in the accurate and effective modeling of the object extension dynamics as well as the centroid kinematics.

Specifically for tracking of an extended object using range extent measurements, the support function fits well with the description of extended object extension [22]. Since the support function has natural and intuitive connections to the range extent of an object, the down-range $D(\theta_k)$ and cross-range extent $C(\theta_k)$ at viewing angle θ_k can be directly expressed by support functions:

$$D(\theta_k) = H(\theta_k) + H(\theta_k + \pi), \quad (5)$$

$$C(\theta_k) = H(\theta_k + \frac{\pi}{2}) + H(\theta_k - \frac{\pi}{2}). \quad (6)$$

However, the extension of a somewhat complex extended object (as an example of Figure 1) in some cases is not feasible or convenient to be described by support functions directly using range extent measurements. This certainly brings barriers to the modeling and estimation of the MCEO extension dynamics and there is no explicit consideration in the literature. The major difficulties are summarized as follows:

- how to accurately describe the extension dynamics (change in size, shape, orientation, e.g., rotation) of an MCEO over time,
- how to deal with the close coupling between the centroid kinematics and extension evolution, and how to embody it in the MCEO modeling.

Remark 1. Compared with traditional maneuvering target tracking, the maneuvering extended object modeling is more difficult and complicated because the extension rotation occurs along with the turn maneuver of the centroid. The close coupling between the centroid maneuver and the change of extension should be considered. Thus, our research on this problem is meaningful and will benefit MCEOT using measurements of range extent, although it is difficult to handle.

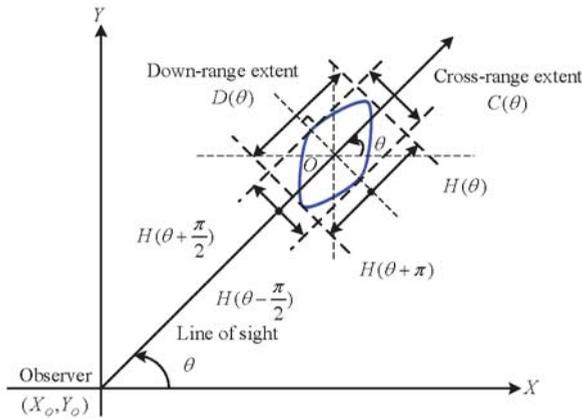


Figure 1. Down-range and cross-range extent.

3. MCEO Modeling Using Range Extent Measurements

3.1. The Unified Complex Extension Dynamics Based on Minkowski Sum

In this section, we first consider decomposing an complex extended object as multiple simple sub-objects by Minkowski sum (also known as Minkowski dilation). In this case, the support function representation of the complex object extension can be easily obtained by merging support function representations of these simple sub-object extensions. This is because the sum of support functions is also a support function by using Minkowski sum [23]. Our study finds that the rotation of the complex object extension occurs along with the simultaneous rotation of multiple simple sub-object extensions, i.e., all their rotation motions have exactly the same rotational model with the same turn rate. As examples of an MCEO in Figure 2, its complex extension dynamics in rotation motion at counterclockwise order can be characterized by modeling extension dynamics of sub-objects jointly. In practical applications, the rotation rate of the object extension and the turn rate of the centroid are exactly the same because the extended object is usually treated as a rigid body. Based on the above analysis, the turn maneuver of a complex extended object can be directly modeled by that of several simple sub-objects via Minkowski sum. That is, the MCEO extension dynamics are characterized by multiple simple sub-objects' extension dynamics jointly, which has a general form as follows:

$$\begin{bmatrix} \mathbf{x}_k^m \\ \mathbf{x}_k^{e,1} \\ \vdots \\ \mathbf{x}_k^{e,n} \end{bmatrix} = \begin{bmatrix} F^m & 0 & 0 & 0 \\ 0 & F^{e,1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & F^{e,n} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1}^m \\ \mathbf{x}_{k-1}^{e,1} \\ \vdots \\ \mathbf{x}_{k-1}^{e,n} \end{bmatrix} + \begin{bmatrix} G^m & 0 & 0 & 0 \\ 0 & G^{e,1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & G^{e,n} \end{bmatrix} \mathbf{w}_{k-1}, \tag{7}$$

where $\mathbf{x}_{k-1}^{e,1}, \dots, \mathbf{x}_{k-1}^{e,n}$ are used to represent each sub-object's extension, and $F^{e,1} \dots F^{e,n}$ denote their respective extension transition matrix. In this way, the MCEO extension dynamics modeling is largely simplified because many simple sub-objects' extension dynamics (e.g., circles, ellipses, rectangles, etc.) are usually available.

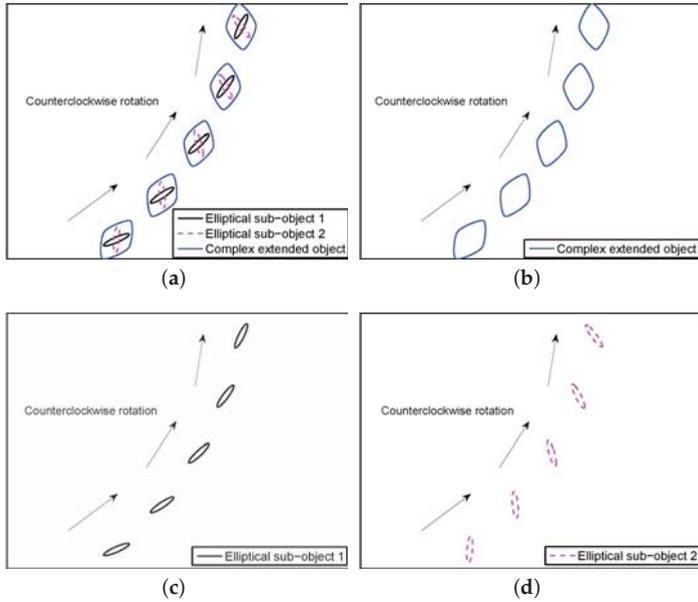


Figure 2. The complex object extension dynamics based on Minkowski sum, (a) illustrative example; (b) extension dynamics of complex object; (c) extension dynamics of sub-object 1; (d) extension dynamics of sub-object 2.

In view of the above, the key to the MCEO modeling based on Minkowski sum for tracking is how to obtain effective and concise forms of the sub-objects’ extension dynamics. As an example of Figure 2, these elliptical sub-objects (decomposed by Minkowski sum) are conveniently described by two 2×2 symmetric positive definite matrices. Thus, we have

$$E_k^1 = \begin{bmatrix} E_k^{(1)} & E_k^{(2)} \\ E_k^{(2)} & E_k^{(3)} \end{bmatrix}, E_k^2 = \begin{bmatrix} E_k^{(4)} & E_k^{(5)} \\ E_k^{(5)} & E_k^{(6)} \end{bmatrix}. \tag{8}$$

For one elliptical sub-object, its support functions representation $H_1(\theta_k)$ at viewing angle θ_k are [22]

$$H_1(\theta_k) = \left(\mathbf{v}_k^T \begin{bmatrix} E_k^{(1)} & E_k^{(2)} \\ E_k^{(2)} & E_k^{(3)} \end{bmatrix} \mathbf{v}_k \right)^{1/2}, \tag{9}$$

where $\mathbf{v}_k = [\cos \theta_k, \sin \theta_k]^T$ is the unit vector. The Matrix entries of E_k^1 can be included as a parameter vector $\mathbf{x}_k^{e,1} = [E_k^{(1)}, E_k^{(2)}, E_k^{(3)}]^T$ for describing the elliptical extension because they carry useful information about this sub-object’s extension (i.e., its geometric properties are fully reflected by them). Thus, this sub-object’s extension state transition is

$$\begin{aligned} \mathbf{x}_k^{e,1} &= \mathbf{x}_{k-1}^{e,1} + \mathbf{w}_{k-1}^{e,1} \\ \Rightarrow \begin{bmatrix} E_k^{(1)} \\ E_k^{(2)} \\ E_k^{(3)} \end{bmatrix} &= \begin{bmatrix} E_{k-1}^{(1)} \\ E_{k-1}^{(2)} \\ E_{k-1}^{(3)} \end{bmatrix} + \begin{bmatrix} w_{k-1}^{e,(1)} \\ w_{k-1}^{e,(2)} \\ w_{k-1}^{e,(3)} \end{bmatrix}, \end{aligned} \tag{10}$$

where the uncertainty of this object extension is embodied by the process noise $w_{k-1}^{e,1} = [w_{k-1}^{e,(1)}, w_{k-1}^{e,(2)}, w_{k-1}^{e,(3)}]^T$. Equation (10) describes the object extension dynamics over time. Actually, this ellipse can be rotated to a different orientation by using A_k , i.e.,

$$\begin{bmatrix} E_k^{(1)} & E_k^{(2)} \\ E_k^{(2)} & E_k^{(3)} \end{bmatrix} = A_k \begin{bmatrix} E_{k-1}^{(1)} + w_{k-1}^{e,(1)} & E_{k-1}^{(2)} + w_{k-1}^{e,(2)} \\ E_{k-1}^{(2)} + w_{k-1}^{e,(2)} & E_{k-1}^{(3)} + w_{k-1}^{e,(3)} \end{bmatrix} A_k^T, \tag{11}$$

which describes the change of extension orientation over time. In a 2D space, every rotation matrix is

$$A_k = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}. \tag{12}$$

Substituting Equation (12) into Equation (11) yields

$$\begin{aligned} \begin{bmatrix} E_k^{(1)} \\ E_k^{(2)} \\ E_k^{(3)} \end{bmatrix} &= \begin{bmatrix} \cos^2 \phi & -\sin 2\phi & \sin^2 \phi \\ \frac{1}{2} \sin 2\phi & \cos 2\phi & -\frac{1}{2} \sin 2\phi \\ \sin^2 \phi & \sin 2\phi & \cos^2 \phi \end{bmatrix} \begin{bmatrix} E_{k-1}^{(1)} \\ E_{k-1}^{(2)} \\ E_{k-1}^{(3)} \end{bmatrix} \\ &+ \begin{bmatrix} \cos^2 \phi & -\sin 2\phi & \sin^2 \phi \\ \frac{1}{2} \sin 2\phi & \cos 2\phi & -\frac{1}{2} \sin 2\phi \\ \sin^2 \phi & \sin 2\phi & \cos^2 \phi \end{bmatrix} \begin{bmatrix} w_{k-1}^{e,(1)} \\ w_{k-1}^{e,(2)} \\ w_{k-1}^{e,(3)} \end{bmatrix} \\ &\implies x_k^{e,1} = F^{e,1} x_{k-1}^{e,1} + G^{e,1} w_{k-1}^{e,1}, \end{aligned} \tag{13}$$

where $\phi = \omega T$ is the rotation angle in counterclockwise order, and

$$F^{e,1} = G^{e,1} = \begin{bmatrix} \cos^2 \omega T & -\sin 2\omega T & \sin^2 \omega T \\ \frac{1}{2} \sin 2\omega T & \cos 2\omega T & -\frac{1}{2} \sin 2\omega T \\ \sin^2 \omega T & \sin 2\omega T & \cos^2 \omega T \end{bmatrix}. \tag{14}$$

For the other elliptical object in Figure 2, its support functions representation $H_2(\theta_k)$ at viewing angle θ_k is

$$H_2(\theta_k) = \left(v_k^T \begin{bmatrix} E_k^{(4)} & E_k^{(5)} \\ E_k^{(5)} & E_k^{(6)} \end{bmatrix} v_k \right)^{1/2}. \tag{15}$$

These matrix entries $E_k^{(4)}, E_k^{(5)}, E_k^{(6)}$ can be taken as its extension parameters $x_k^{e,2} = [E_k^{(4)}, E_k^{(5)}, E_k^{(6)}]^T$. Since the rotation motion of two sub-objects have exactly the same rotation mode with the same turn rate, the sub-object extension transition of $x_k^{e,2}$ is

$$x_k^{e,2} = F^{e,1} x_{k-1}^{e,2} + G^{e,1} w_{k-1}^{e,2}, \tag{16}$$

where $w_{k-1}^{e,2} = [w_{k-1}^{e,(4)}, w_{k-1}^{e,(5)}, w_{k-1}^{e,(6)}]^T$. As mentioned before, the extension dynamics of this complex extended object in rotation motion can be characterized by modeling extension dynamics of sub-objects jointly from Equations (13) and (16), i.e.,

$$\begin{aligned} \begin{bmatrix} x_k^{e,1} \\ x_k^{e,2} \end{bmatrix} &= \begin{bmatrix} F^{e,1} & 0 \\ 0 & F^{e,1} \end{bmatrix} \begin{bmatrix} x_{k-1}^{e,1} \\ x_{k-1}^{e,2} \end{bmatrix} + \begin{bmatrix} G^{e,1} & 0 \\ 0 & G^{e,1} \end{bmatrix} \begin{bmatrix} w_{k-1}^{e,1} \\ w_{k-1}^{e,2} \end{bmatrix}, \\ x_k^e &= F^e x_{k-1}^e + G^e w_{k-1}^e, \end{aligned} \tag{17}$$

where $F^e = G^e = \text{diag}(F^{e,1}, F^{e,1}) = \text{diag}(G^{e,1}, G^{e,1})$. These elliptical sub-object have the same extension evolution over time for describing the turn maneuver, which facilitates modeling of the complex object extension dynamics for tracking.

Note that other complex object extensions can also be described as the Minkowski sums of other simple sub-objects with different extensions (besides ellipses) represented by support functions. The elliptical maneuvering object model is a by-product and actually considered as a special case of the proposed approach based on Minkowski sum.

Remark 2. $F^e = G^e = \text{diag}(1, 1, 1, 1, 1, 1)$ in Equation (17) when the turn rate $\omega = 0$. Correspondingly, the centroid kinematic state transition matrix of the CT motion in Equation (3) is also changed to $F^m = \text{diag}(F^{CV}, F^{CV})$, where $F^{CV} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$. The centroid state transition in Equation (4) is rewritten by

$$x_k^m = \text{diag}(F^{CV}, F^{CV})x_{k-1}^m + G^m w_{k-1}^m. \tag{18}$$

In this case, the proposed maneuvering extended object model for MCEOT (e.g., describing constant turn motion) reduces to the non-maneuvering object model (e.g., describing constant velocity motion) when the turn rate $\omega = 0$. Given more information, we can design more specifically for different MCEOT scenarios because different F^m and F^e result in different centroid state and object extension transition, respectively.

As mentioned above, there is a close coupling between the centroid maneuver and the change of extension, i.e., the extension rotation occurs along with the turn maneuver of the centroid. Thus, the rotation rate of extension $\omega = \frac{\dot{\phi}}{T}$ in A_k (see Equation (12)) and the turn rate ω of the centroid (see Equation (3)) are exactly the same when an extended object maneuvers (e.g., performing a turn motion). The transition of x_k^e in Equation (17) is actually the object extension dynamic model, which describes the change of the extension orientation over time. Equation (7) also explicitly describes the relation between the object extension orientation change and the centroid turn maneuver. The extension state $x_k^e = [(x_k^{e,1})^T, (x_k^{e,2})^T]^T$ is included in $x_k = [x_k, \dot{x}_k, y_k, \dot{y}_k, E_k^{(1)}, E_k^{(2)}, E_k^{(3)}, E_k^{(4)}, E_k^{(5)}, E_k^{(6)}]^T$ as the joint state vector. Note that different F^m and F^e describe different transition of centroid kinematic state and the change of object extension orientation, respectively.

Remark 3. The MCEO modeling approach based on Minkowski sum using range extent measurements has several following advantages: (1) It has a concise mathematical form to describe an MCEO with different turn motions accurately; (2) the close relationship between the maneuver of the centroid and the change of object extension orientation is considered and solved explicitly; (3) the proposed modeling approach is easy to implement and facilitates the derivation and design of MCEOT algorithms; and (4) the matrix parameterization is not the only option for our approach to solve the MCEOT using target’s range extent measurements. Other parameterizations can also be adopted in the unified model framework based on Minkowski sum if they are efficient to describe other convex objects. This does not affect the generality of the proposed approaches.

For tracking of an MCEO (e.g., a maneuvering civil or military aircraft) that performs CT motions with known/unknown turn rates, we propose two different Minkowski-sum-based MCEOT algorithms for estimating the joint target state x_k (i.e., both the centroid kinematic state x_k^m and the extension state x_k^e) in this section.

3.2. The Minkowski-Sum-Based Modeling and Estimation for CT Maneuvers with Known Turn Rates

In the first case of constant turn maneuvers with known turn rates, we introduce a hybrid system framework for describing practical motions and extension dynamics of an extended object accurately. It is beneficial to use more than one MCEO motion model with known turn rates in a tracking algorithm when the true object motion is complicated, e.g., the whole process of turn motions is assumed to

be described by a model set (e.g., including a non-maneuvering model and several turn maneuver models). Thus, we consider the following hybrid system:

$$\begin{aligned} \mathbf{x}_k &= F^{(i)} \mathbf{x}_{k-1} + \mathbf{w}_{k-1}^{(i)}, \mathbf{w}_{k-1}^{(i)} \sim \mathcal{N}(0, \mathbf{Q}_k^{(i)}) \\ \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{v}_k), \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k), \end{aligned} \tag{19}$$

where superscript i denotes quantities pertinent to model $m^{(i)}$ in the set $\mathcal{M} = \{m^{(1)}, m^{(2)}, \dots, m^{(M)}\}$. The modal state m_k sequence is a first-order Markov chain that has transition probabilities $\pi_{i,j} = P\{m_k^{(j)} | m_{k-1}^{(i)}\}, \forall m^{(i)}, m^{(j)}, k$. $m_k^{(i)}$ means that model $m^{(i)}$ matching the system motion mode (e.g., turn motions with different turn rates) is in effect at time k .

Similar to [17,20], we also assume that a high resolution radar provides target range extent measurements (i.e., $D(\theta_k)$ and $C(\theta_k)$) as well as the range r_k and bearing β_k measurements of the object centroid. Thus, the measurement equation is written as

$$\mathbf{z}_k = [r_k, \beta_k, D_k, C_k]^T + \mathbf{v}_k, \tag{20}$$

where $r_k = \sqrt{(x_k - X_0)^2 + (y_k - Y_0)^2}$, $\beta_k = \arctan\left(\frac{y_k - Y_0}{x_k - X_0}\right)$ and \mathbf{v}_k is measurement noise. (X_0, Y_0) is the location of the sensor. The measurements in $\mathbf{z}_k = [r_k, \beta_k, D_k, C_k]^T$ are usually provided from different physical channels, and the noise \mathbf{v}_k is generally assumed to be a zero-mean Gaussian process with $\text{cov}[\mathbf{v}_k] = \mathbf{R}_k = \text{diag}[R_k^r, R_k^\beta, R_k^D, R_k^C]$. As an example of a complex object in Figure 1, this object can be described as a Minkowski sum of two elliptical sub-objects, and its support function representation has the following form: $H(\theta_k) = H_1(\theta_k) + H_2(\theta_k)$. Thus, Equations (5) and (6) are rewritten as

$$D(\theta_k) = H(\theta_k) + H(\theta_k + \pi) = H_1(\theta_k) + H_2(\theta_k) + H_1(\theta_k + \pi) + H_2(\theta_k + \pi), \tag{21}$$

$$C(\theta_k) = H(\theta_k + \frac{\pi}{2}) + H(\theta_k - \frac{\pi}{2}) = H_1(\theta_k + \frac{\pi}{2}) + H_2(\theta_k + \frac{\pi}{2}) + H_1(\theta_k - \frac{\pi}{2}) + H_2(\theta_k - \frac{\pi}{2}). \tag{22}$$

Since the elliptical sub-objects are centrosymmetric, i.e.,

$$H(\theta_k) = H(\theta_k + \pi), H(\theta_k + \frac{\pi}{2}) = H(\theta_k - \frac{\pi}{2}), \tag{23}$$

we have

$$D(\theta_k) = 2H(\theta_k) = 2(H_1(\theta_k) + H_2(\theta_k)), \tag{24}$$

$$C(\theta_k) = 2H(\theta_k + \frac{\pi}{2}) = 2(H_1(\theta_k + \frac{\pi}{2}) + H_2(\theta_k + \frac{\pi}{2})). \tag{25}$$

For this case of turn maneuvers with known turn rates, we propose an MCEOT algorithm within the multiple model framework [26] by comprehensively considering the uncertainties of motion and extension dynamics. It runs a set of filters based on models describing several possible turn maneuvers as well as the non-maneuver target motion. Suppose that the required state estimate $\hat{\mathbf{x}}_{k-1|k-1}^{(i)} = [\hat{\mathbf{x}}_{k-1|k-1}^{m(i)}, \hat{\mathbf{x}}_{k-1|k-1}^{e(i)}]^T$, its error covariance $\bar{P}_{k-1|k-1}^{(i)}$ and model probability $\mu_{k-1}^{(i)}$ at time $k-1$ are available for $m^{(i)}, i = 1, \dots, N$. $\hat{\mathbf{x}}_{k|k}$ and $P_{k|k}$ can be obtained recursively ($k-1 \rightarrow k$) in prediction and updated as follows.

Step 1. Evaluate the mixing probabilities $\mu_{k-1}^{j|i} = \pi_{j,i} \mu_{k-1}^{(j)} / \mu_{k|k-1}^{(i)}$ with $\mu_{k|k-1}^{(i)} = \sum_{j=1}^M \pi_{j,i} \mu_{k-1}^{(j)}$, and mixing estimates $\hat{\mathbf{x}}_{k-1|k-1}^{(i)} = \sum_{j=1}^M \hat{\mathbf{x}}_{k-1|k-1}^{(j)} \mu_{k-1}^{j|i}$ and covariance $\bar{P}_{k-1|k-1}^{(i)} = \sum_{j=1}^M [P_{k-1|k-1}^{(j)} + (\hat{\mathbf{x}}_{k-1|k-1}^{(i)} - \hat{\mathbf{x}}_{k-1|k-1}^{(j)})(\hat{\mathbf{x}}_{k-1|k-1}^{(i)} - \hat{\mathbf{x}}_{k-1|k-1}^{(j)})^T] \mu_{k-1}^{j|i}$.

Step 2. Run a nonlinear filter (e.g., unscented filter, extended Kalman filter, quadrature Kalman filter, etc.) for each model with initial condition $(\hat{\mathbf{x}}_{k-1|k-1}^{(i)} = [\hat{\mathbf{x}}_{k-1|k-1}^{m(i)}, \hat{\mathbf{e}}_{k-1|k-1}^{(i)}]^T, P_{k-1|k-1}^{(i)}$ and $\mu_{k-1}^{(i)})$.

Step 3. Model probability update is derived by Bayes' formula, i.e., $\mu_k^{(i)} = \frac{\Lambda_k^{(i)} \mu_{k|k-1}^{(i)}}{\sum_{j=1}^N \Lambda_k^{(j)} \mu_{k|k-1}^{(j)}}$, where $\Lambda_k^{(i)} = p[z_k|z^{k-1}, m_k^{(i)}]$ is the likelihood of model $m_k^{(i)}$. It is usually approximated as a Gaussian distribution by moment matching, i.e. $\Lambda_k^{(i)} = \mathcal{N}(z_k - \hat{z}_{k|k-1}^{(i)}; 0, S_k^{(i)})$.

Step 4. The fused kinematic state and extension state estimate is calculated as the sum of $\hat{\mathbf{x}}_{k|k}^{(i)} = [\hat{\mathbf{x}}_{k|k}^{m,(i)}, \hat{\mathbf{e}}_{k|k}^{(i)}]^T$ weighted by its corresponding model probabilities $\mu_k^{(i)}$: $\hat{\mathbf{x}}_{k|k} = \sum_{i=1}^N \hat{\mathbf{x}}_{k|k}^{(i)} \mu_k^{(i)}$ and $P_{k|k} = \sum_{i=1}^N [P_{k|k}^{(i)} + (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k}^{(i)})(\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k}^{(i)})^T] \mu_k^{(i)}$.

Remark 4. Thanks to the concise linear form of the Minkowski-sum-based dynamic model given the known turn rate, the uncertainty in turn rate can be alleviated by the proposed algorithm. However, turn maneuvers are not necessarily totally covered by several CT models with known turn rates—not to say the true turn rate is usually unknown for the tracker in practical applications. This case can be handled by another modeling and estimation algorithm proposed next.

3.3. The Minkowski-Sum-Based Modeling and Estimation for CT Maneuvers with Unknown Turn Rates

For tracking of an MCEO that performs CT motions with unknown turn rates, we propose another Minkowski-sum-based MCEO modeling and estimation algorithm. In many CT maneuver cases, the turn rate is usually unknown or not be known a priori for the tracker. Thus, different from the case of constant turn maneuvers with known turn rates, it can be described by the Wiener process:

$$\omega_k = \omega_{k-1} + \mathbf{w}_{k-1}^\omega \tag{26}$$

where \mathbf{w}_{k-1}^ω is zero-mean Gaussian noise. In this case, the unknown turn rate is included as a state component into the joint state vector, to be estimated recursively. That is, we augment the object state vector $\mathbf{x}_k = [(\mathbf{x}_k^m)^T, (\mathbf{x}_k^e)^T]^T$ to include it:

$$\mathbf{x}_k^A \triangleq [(\mathbf{x}_k)^T, \omega_k]^T \tag{27}$$

The MEO dynamic model of this case is reformulated by Equations (2) and (26) jointly, i.e.,

$$\begin{bmatrix} \mathbf{x}_k^m \\ \mathbf{x}_k^e \\ \omega_k \end{bmatrix} = \begin{bmatrix} F_{k-1}^m & 0 & 0 \\ 0 & F_{k-1}^e & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1}^m \\ \mathbf{x}_{k-1}^e \\ \omega_{k-1} \end{bmatrix} + \begin{bmatrix} G_{k-1}^m & 0 & 0 \\ 0 & G_{k-1}^e & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{w}_{k-1}^m \\ \mathbf{w}_{k-1}^e \\ \mathbf{w}_{k-1}^\omega \end{bmatrix}, \tag{28}$$

where

$$F_{k-1}^m = \begin{bmatrix} 1 & \frac{\sin(\omega_{k-1}T)}{\omega_{k-1}} & 0 & -\frac{1-\cos(\omega_{k-1}T)}{\omega_{k-1}} \\ 0 & \cos(\omega_{k-1}T) & 0 & -\sin(\omega_{k-1}T) \\ 0 & \frac{1-\cos(\omega_{k-1}T)}{\omega_{k-1}} & 1 & \frac{\sin(\omega_{k-1}T)}{\omega_{k-1}} \\ 0 & \sin(\omega_{k-1}T) & 0 & \cos(\omega_{k-1}T) \end{bmatrix}, \tag{29}$$

and $F_{k-1}^e = G_{k-1}^e = \text{diag}(F_{k-1}^{e,1}, F_{k-1}^{e,1})$,

$$F_{k-1}^{e,1} = \begin{bmatrix} \cos^2 \omega_{k-1}T & -\sin 2\omega_{k-1}T & \sin^2 \omega_{k-1}T \\ \frac{1}{2} \sin 2\omega_{k-1}T & \cos 2\omega_{k-1}T & -\frac{1}{2} \sin 2\omega_{k-1}T \\ \sin^2 \omega_{k-1}T & \sin 2\omega_{k-1}T & \cos^2 \omega_{k-1}T \end{bmatrix}. \tag{30}$$

Thus, the following dynamic and measurement equations of the system are considered:

$$\begin{aligned} \mathbf{x}_k^A &= F_{k-1}^A \mathbf{x}_{k-1}^A + G_{k-1}^A \mathbf{w}_{k-1}^A, \quad \mathbf{w}_{k-1}^A \sim \mathcal{N}(0, \mathbf{Q}_k^A) \\ \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{v}_k), \quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k) \end{aligned} \tag{31}$$

Note that $F_{k-1}^A = \text{diag}(F_{k-1}^m, F_{k-1}^e, 1)$ in Equation (31) is totally different from $F = \text{diag}(F^m, F^e)$ in Equation (19) because here the turn rate ω_{k-1} is considered as an unknown state component. In this case, the dynamic model of Equation (28) is really no longer linear, and a nonlinear tracking algorithm needs to be designed with a joint state vector x_k^A (including the unknown turn rate). The measurement equation of Equation (31) is still the same as those of Equation (19). Suppose that augmented state estimate $\hat{x}_{k-1|k-1}^A = [(\hat{x}_{k-1|k-1}^A)^T, \omega_k]^T$ and its error covariance $P_{k-1|k-1}^A$ at time $k - 1$ are available, we adopt the unscented transformation (UT) [27] to solve the severely nonlinear dynamic and measurement equations, i.e.,

$$\begin{aligned} (\hat{x}_{k|k-1}^A, P_{k|k-1}^A) &= \text{UT}[F_{k-1}^A x_{k-1}^A, (\hat{x}_{k-1|k-1}^A)^T, P_{k-1|k-1}^A], \\ (\hat{z}_{k|k-1}, S_k) &= \text{UT}[h(x_k^A), (\hat{x}_{k|k-1}^A)^T, P_{k|k-1}^A]. \end{aligned} \tag{32}$$

The UT is an effective approach to approximating first and second moments (e.g., mean and covariance) as a nonlinear function of random state vectors x_k^A and z_k by deterministic sample points with weights $\{a^i, i = 0, 1, \dots, N\}$. Note that the UT is not the only option; other moment approximating methods (e.g., the uncorrelated conversion [28,29], the Gauss–Hermite quadrature rules [30], the Cubature rules [31], etc.) may also be used here. Excluding Equation (32), the remaining estimate process of Equation (31) can be implemented in the framework of Kalman filter directly due to the proposed concise model describing CT maneuvers with unknown turn rates.

3.4. Complexity Analysis

In our modeling and tracking framework, a complex object is decomposed as multiple simple sub-objects (i.e., K_1, K_2, \dots, K_{N_s}) by using the Minkowski sum. If they are represented by support functions (i.e., $H_{K_1}(\theta), H_{K_2}(\theta), \dots, H_{K_{N_s}}(\theta)$) for the viewing angle θ , the Minkowski sum $K_1 \oplus K_2 \oplus \dots \oplus K_{N_s}$ is unique with

$$H_{K_1 \oplus K_2 \oplus \dots \oplus K_{N_s}}(\theta) = H_{K_1}(\theta) + H_{K_2}(\theta) + \dots + H_{K_{N_s}}(\theta), \tag{33}$$

where $H_{K_1}(\theta) + H_{K_2}(\theta) + \dots + H_{K_{N_s}}(\theta)$ is also a support function. As an example of Figure 2, the Minkowski sum of a complex object (i.e., $K_1 \oplus K_2$) becomes $H_{K_1 \oplus K_2}(\theta) = H_{K_1}(\theta) + H_{K_2}(\theta)$. Correspondingly, its extension state $x_k^e = [E_k^{(1)}, E_k^{(2)}, E_k^{(3)}, E_k^{(4)}, E_k^{(5)}, E_k^{(6)}]^T$ consists of the subjects' extension state $x_k^{e,1} = [E_k^{(1)}, E_k^{(2)}, E_k^{(3)}]^T$ and $x_k^{e,2} = [E_k^{(4)}, E_k^{(5)}, E_k^{(6)}]^T$.

As mentioned in Sections 3.2 and 3.3, the extension state is estimated by calculating the needed moments of x_k^e and P_k^e based on UT sampling, and implemented in the framework of Kalman filter (KF). The general KF has $\mathcal{O}(N^3)$ complexity in the number N of state and requires $\mathcal{O}(N^3)$ floating-point computations [32]. Thus, the computational complexity of our approach has the same order of magnitude as that of KF and mainly depends on the dimension of the selected extension state for tracking. As an example of an elliptical object, its computational complexity is $\mathcal{O}(N^3)$. Since one elliptical object's extension state is $x_k^{e,1} = [E_k^{(1)}, E_k^{(2)}, E_k^{(3)}]^T$ with the dimension of 3, the computational complexity of a complex object is $\mathcal{O}(N^3 + [3(N_s - 1)]^3)$, where N_s is the number of the decomposed elliptical sub-objects. Actually, $\mathcal{O}(N^3 + [3(N_s - 1)]^3)$ is equivalent to $\mathcal{O}(N^3) + \mathcal{O}([3(N_s - 1)]^3)$ by neglecting the inconsequential high order items in practical applications. In other words, compared with the computational complexity of a single sub-object's extension state estimation, the complex object's computational complexity is added by $\mathcal{O}([3(N_s - 1)]^3)$. The above will be analyzed and verified by the following experimental simulations in Section 4.3.

4. Simulation Results and Performance Evaluation

In this section, we focus on the case of a stationary sensor platform and moving extended object. Here, the proposed Minkowski-sum-based maneuvering modeling is applied to the MCEOT using range extent measurements, and the following three simulation scenarios of MCEOT are considered.

In the first two scenarios, the proposed MCEO model is a constant turn maneuver model with known turn rates and its dynamic equation is linear. In the third scenario, it is a constant turn model with unknown turn rates and its dynamic equation is nonlinear.

To evaluate the tracking performance of the proposed approach, the following two approaches are compared by simulation to illustrate its effectiveness in these scenarios. To be fair for extension estimation, the compared approaches are initialized with the same extension (e.g., a circle) without further information:

- (a) MCEOT-1: The proposed approach based on Minkowski sum considering the highly coupled dynamics of both the state and the extension.
- (b) MCEOT-2: The approach considering only the centroid state dynamics.

MCEOT-2 is compared because there is rarely accounted for in existing literature to deal with the MCEOT of both centroid kinematics and the extension dynamics using target's range extent. Thus, other techniques are not compared with the proposed approach (i.e., MCEOT-1) in simulations. However, to verify its effectiveness and benefit of considering the highly coupled dynamics of both the state and the extension for modeling and tracking, we compare MCEOT-1 with MCEOT-2 in the simulations.

The performance comparison results demonstrate that MCEOT-1 outperforms MCEOT-2, in which exchange of information between centroid kinematics and extension dynamics are sufficiently utilized to improve performance. The detailed discussions and analysis are as follows.

4.1. Tracking Performance Using Minkowski-Sum-Based CT Model with Known Rates

Consider the following two different scenarios in which an object with complex extension (as an example of Figure 1) performs a nearly constant velocity (CV) motion and two CT maneuvers in the 2D Cartesian coordinate system with the initial kinematic state $x_0^m = [1000\text{ m}, 100\text{ m/s}, 2000\text{ m}, 60\text{ m/s}]^T$. This object is composed and modeled by two elliptical sub-objects with different orientation angles using Minkowski sum, both of which have the same object extension (i.e., the lengths of minor and major axes are 10 m and 50 m, respectively). The sensor, fixed at the origin (0,0), provides measurements of range, bearing, down- and cross-range extent along line of sight every $T = 1$ s. Each measurement is corrupted by zero-mean white Gaussian noise with standard deviations $\sigma_r = 5$ m, $\sigma_\beta = 0.01$ rad, $\sigma_D = 5$ m, and $\sigma_C = 5$ m. These two scenarios are simulated to illustrate the effectiveness of our approach.

In the scenario A, the true trajectory is illustrated in Figure 3, i.e., this object undergoes two different turn maneuvers. The modal state of this scenario contains $M = 3$ elements for the CV model and two CT models with turn rates $\omega_A^{(1)} = 5\pi/180$ rad/s and $\omega_A^{(2)} = 10\pi/180$ rad/s (it has a stronger maneuverability).

In scenario B, the true trajectory (illustrated in Figure 4) is more sophisticated, in which this object performs different turn motions in clockwise and anticlockwise order. The modal state of scenario B also contains $M = 3$ elements for the CV model and two CT models with turn rates $\omega_B^{(1)} = -5\pi/180$ rad/s and $\omega_B^{(2)} = 10\pi/180$ rad/s.

Simulation results for the above scenarios are shown in Figures 3–6. The true and estimated object trajectories of different scenarios are shown in Figures 3 and 4. In this simulation, the comparison results of kinematic state estimation are the root-mean-square error (RMSE) [33], i.e., it is chosen as the measure to evaluate performance of position and velocity estimation over 100 Monte Carlo runs ($M = 100$). The RMSE of kinematics estimation has the following forms:

$$\text{RMSE}(\hat{x}_k^m) = \left(\frac{1}{M} \sum_{i=1}^M \|\tilde{x}_{k,i}^m\|^2 \right)^{1/2}, \quad (34)$$

where $\tilde{x}_{k,i}^m = x_{k,i}^m - \hat{x}_{k,i}^m$ is the estimation error on the i th of the M Monte Carlo runs. The comparison results are shown in Figures 5a,b and 6a,b. These figures show that MCEOT-1 has better performance

than MCEOT-2 in estimation of kinematic state because MCEOT-2 considering only the state dynamics. It can be concluded that sufficiently utilizing extension dynamics information of the maneuvering extended object can effectively improve the accuracy of the kinematic state estimation. Actually, with the help of kinematic state estimation, the extension estimation can also be improved.

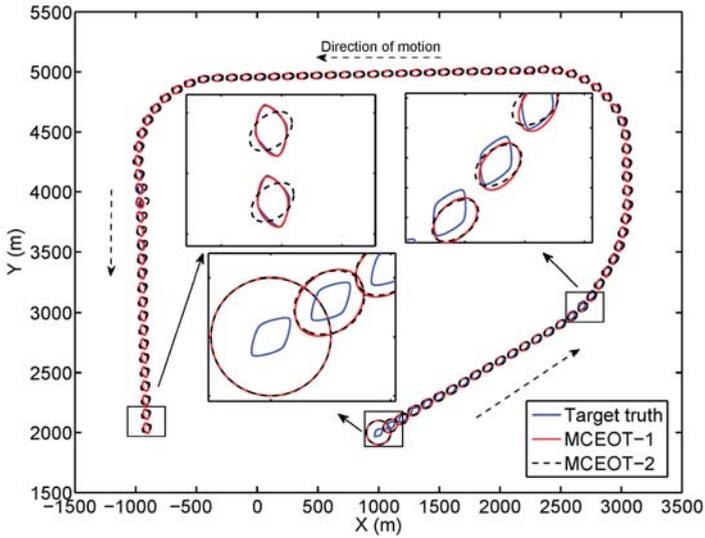


Figure 3. Trajectory of the complex extended object in scenario A (the blue solid line is for the true object, the red solid line and black dash line are for MCEOT-1 and MCEOT-2, respectively).

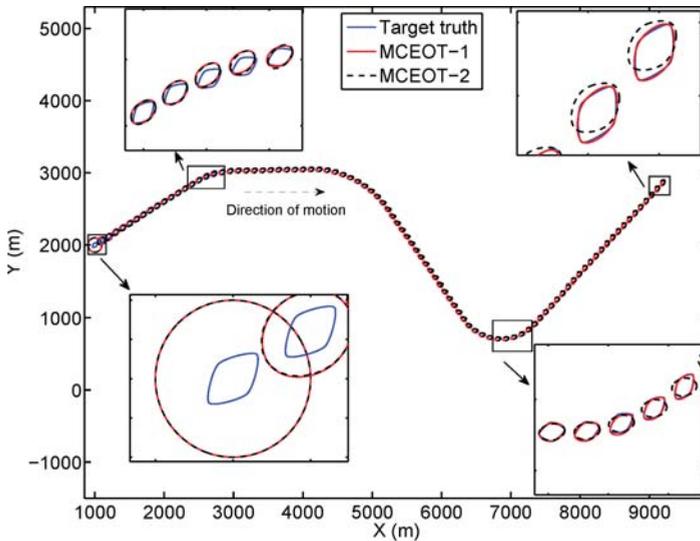


Figure 4. Trajectory of the complex extended object in scenario B (the blue solid line is for the true object, the red solid line and black dash line are for MCEOT-1 and MCEOT-2, respectively).

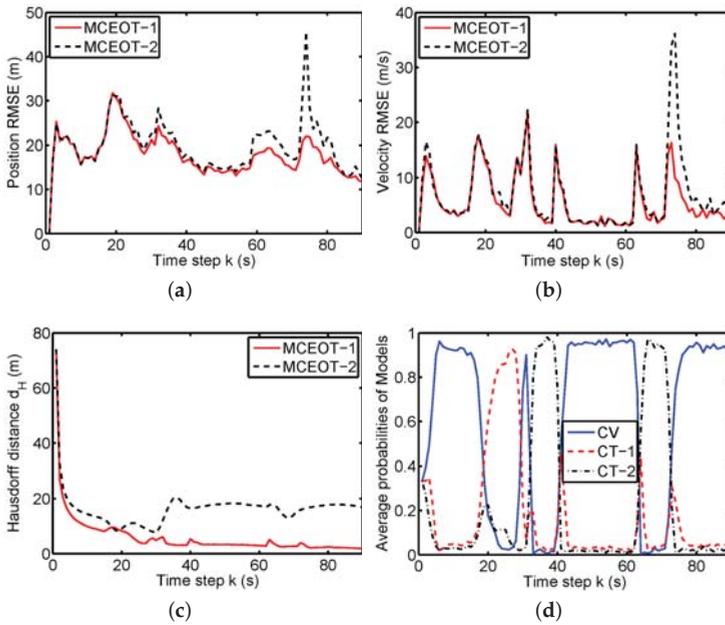


Figure 5. Performance comparison for scenario A. (a) position RMSE; (b) velocity RMSE; (c) Hausdorff distance; (d) average probability of MCEOT-1.

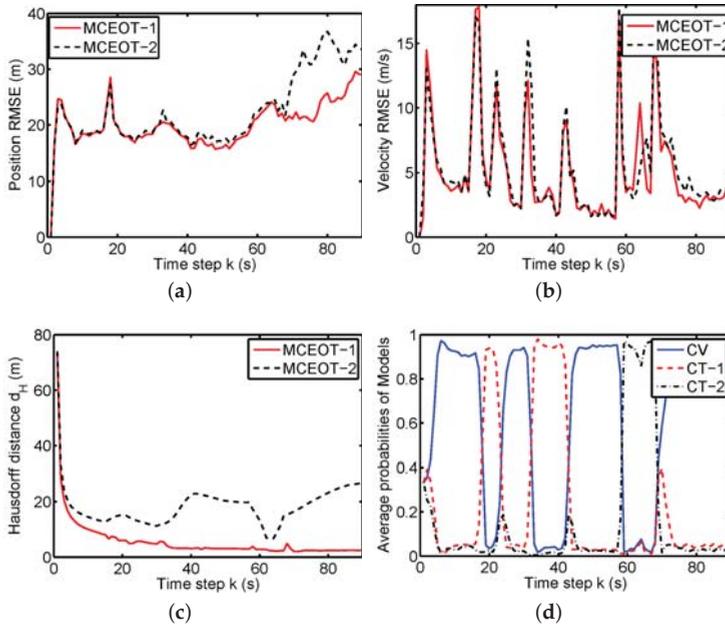


Figure 6. Performance comparison for scenario B. (a) position RMSE; (b) velocity RMSE; (c) Hausdorff distance; (d) average probability of MCEOT-1.

The object extension evaluation of an maneuvering extended object can be regarded as a problem of object extension matching; thus, the Hausdorff distance [34] is adopted to objectively measure the quality of extension estimation, which reflects the degree of similarity between the estimated extension and the true one. However, it is difficult to be used for performance evaluation of the extension estimation directly because the range of θ is a continuous set (i.e., $\theta \in [0, 2\pi)$). Thus, the Hausdorff distance is modified by

$$d_H(K, \hat{K}) = \frac{1}{M} \sum_{i=1}^M \sup_{\theta \in \{j \cdot \frac{2\pi}{N_s}, j=1, \dots, N_s\}} |H_{K,i}(\theta) - H_{\hat{K},i}(\theta)|, \tag{35}$$

where $d_H(K, \hat{K})$ is the modified Hausdorff distance between the estimated object extension \hat{K} and the true one K on the i th of the M Monte Carlo runs, and $\theta \in [0, 2\pi)$ is replaced by the discrete set $\theta \in \{j \cdot \frac{2\pi}{N_s}, i = 1, \dots, N_s\}$ via uniform angle sampling. In our simulation, N_s as the number of sampling points is set to be 1000.

From Figures 3–6, we can see that MCEOT-1 has almost the same performance as MCEOT-2 in the centroid kinematics and extension estimation when the extended object performs CV motion. However, MCEOT-1 outperforms MCEOT-2 (i.e., MCEOT-1 has smaller position and velocity RMSE, and the short Hausdorff distance) when this object maneuvers. In other words, MCEOT-1 estimates the kinematic state and object extension accurately while MCEOT-2 cannot (see Figures 5a–c and 6a–c) because MCEOT-2 only focuses on centroid kinematics without considering of the object extension dynamics. Compared with MCOET-2, MCEOT-1 can produce simultaneously stable tracking and extension estimates converging to the true extended object.

As illustrated in Figures 5d and 6d, MCEOT-1 distinguishes the true mode successfully and identifies the tracking model correctly matches the real situation, especially when the extended object starts or ends a maneuver.

4.2. Tracking Performance Using Minkowski-Sum-Based CT Model with Unknown Rates

In the scenario C, the true trajectory is illustrated in Figure 7a: the extended object performs CT motion with turn rate $\omega_C = \pi/180$ from 1 s to 60 s. Unlike in Section 4.1, the turn rate ω_C of CT motion is totally unknown for the tracker in this scenario. Note that the other simulation parameters are the same as that of the extended objects in the Section 4.1. In addition, the compared approaches (i.e., MCEOT-1 and MCEOT-2) are also initialized with the same circle without further extension information for the fair evaluation.

From Figure 7b–d, we can see that simulation results and performance comparisons in this case are similar to that of Section 4.1, i.e., MCEOT-1 has better performance than MCEOT-2 in the centroid kinematics and extension estimation. Note that MCEOT-2 outperforms MCEOT-1 in the initial tracking process (roughly from 1 s to 20 s) because MCEOT-2 considers only the centroid state dynamics without extension dynamics, i.e., its dynamic model is:

$$\begin{bmatrix} \mathbf{x}_k^m \\ \mathbf{e}_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} F_{k-1}^m & 0 & 0 \\ 0 & I_6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1}^m \\ \mathbf{e}_{k-1} \\ \omega_{k-1} \end{bmatrix} + \begin{bmatrix} G_{k-1}^A & 0 & 0 \\ 0 & I_6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{w}_{k-1}^m \\ \mathbf{w}_{k-1}^e \\ \mathbf{w}_{k-1}^\omega \end{bmatrix}, \tag{36}$$

where $I_6 = \text{diag}(1, 1, 1, 1, 1, 1)$. This dynamic model is really nonlinear because this unknown turn rate is considered as a unknown state component and augmented by the joint state. However, compared with the dynamic model of Equation (28), its nonlinearity is weaker for tracking. Thus, MCEOT-2 achieves better performance than MCEOT-1 in the initial period. As time goes on, however, MCEOT-1 beats MCEOT-2 in the final steady state (see Figure 7). This main reason is that MCEOT-1 considers the highly coupled dynamics of both the state and the extension. The estimation of the kinematic state and extension are inter-dependent and they affect each other. In this case, more and more two-way

exchange of information between centroid kinematics and extension dynamics is sufficiently utilized by MCEOT-1 to achieve much better estimation performance than MCEOT-2.

The above simulation results demonstrate that good solutions to the MCEOT problem need to fully consider the highly coupled dynamics of both the state and the extension. It can be concluded that the validity of the proposed approach (MCEOT-1) is verified by simulation results and performance comparison in different scenarios.

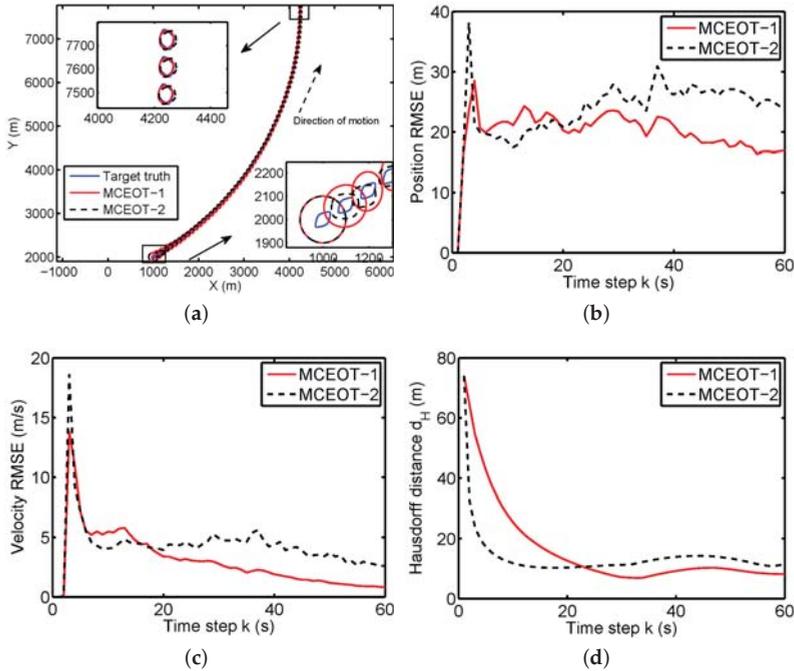


Figure 7. Simulation results in scenario C. (a) the object trajectory; (b) position RMSE; (c) velocity RMSE; (d) Hausdorff distance.

4.3. Performance Comparison and Complexity Analysis

To verify the complexity analysis of Section 3.4 in the experimental aspect, the following approach is compared with MCEOT-1 in scenario A.

MEOT-1: The elliptical maneuvering extended object modeling and tracking approach.

Note that the simulation parameters are the same as that of the Section 4.1, and the compared approaches are still initialized with the same extension (e.g., a circle) without further information. The computational complexities of the MCEOT-1 and the MEOT-1 are analyzed and compared as follows. Specifically for MCEOT-1 and MEOT-1, their computational complexities are reflected by the time consumption in simulations. The most time-consuming parts of the above approaches lie in the estimation of the object extension state. MCEOT-1 has more complexity mainly because it requires more computational time than MEOT-1 for estimating the extension state with higher dimension. Table 1 shows the one-run computational time of the compared approaches for scenarios A and B averaged over the 100 Monte Carlo runs. In the simulation experiments, the above two approaches are implemented in the Matlab (R2011b) environment on a computer with a 4.00 GHz CPU (Intel Core i7 4790k), only a single thread is used). Clearly from this table, MCEOT-1 requires more computational time for tracking (aiming at estimating $x_k^e = [(x_k^{e,A})^T, (x_k^{e,B})^T]^T = [E_k^{(1)}, E_k^{(2)}, E_k^{(3)}, E_k^{(4)}, E_k^{(5)}, E_k^{(6)}]^T$) than

MEOT-1 (aiming at estimating $x_k^{e,A} = [E_k^{(1)}, E_k^{(2)}, E_k^{(3)}]^T$). The results of Table 1 are consistent with the complexity analysis in Section 3.4.

Table 1. Averaged computation time (seconds) for one run (90 steps) of two approaches in scenario A.

MCOET-1	MEOT-1
0.2971	0.1103

As shown in Figure 8, MCEOT-1 outperforms MEOT-1 in estimation of extension state for scenario A. The main reason is that using the simple elliptical extension dynamics model for tracking of a complex object ignores more detailed shape information. This does not facilitate the improvement of estimation performance. Specially during the extended object maneuvers, MCEOT-1 can estimate the object extension accurately while MEOT-1 cannot because the MEOT-1 does not consider the actual extension maneuvers and uses an inappropriate model. As time goes by, more and more tracking errors are cumulated, which go against the accurate estimation of a maneuvering complex extended object. Correspondingly, it achieves much weaker performance than MCEOT-1 in the final steady state (see Figure 8), though it takes less time.

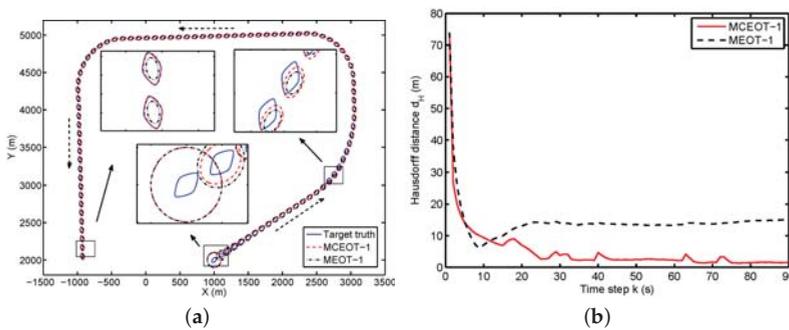


Figure 8. Simulation results in scenario A. (a) the object trajectory; (b) Hausdorff distance.

In summary, for both MCEOT-1 and MEOT-1, their computational complexities mainly differ in the estimation of the object extension state, which are reflected by the time-consumption of tracking. Compared with MEOT-1, MCEOT-1 gets much better results and its time complexity is of the same order. Due to the concise linear form, the proposed approach can achieve excellent tracking performance with the rise of low complexity.

5. Conclusions

To deal with MCEOT using range extent measurements, this paper proposed a general approach based on Minkowski sum. It not only accurately describes different turn maneuvers in both the extension dynamics and the centroid kinematics, but also fully considers the close coupling between the centroid maneuver and the abrupt change of extension. Thank to its concise and unified form, the complex extension dynamics can be easily described by several sub-objects' extension dynamics in a joint way. This largely simplifies the whole modeling process of an MCEO. Furthermore, the proposed Minkowski-sum-based modeling is simple and effective, and it can be easily implemented to deal with different cases of the turn maneuvers with known/unknown turn rates for MCEOT. The effectiveness of what we proposed is demonstrated through simulation results, which achieves excellent performance in the estimation of the centroid kinematic state and the object extension jointly with low complexity.

Furthermore, the practical benefits of our approach are summarized as follows: (1) due to its concise mathematical forms and favorable properties, the maneuvering complex extended object modeling is largely simplified and more detailed shape information is described; (2) a larger range of maneuvering complex extended object tracking can be handled by using Minkowski sum. In summary, our approach may pave the way for solving different MCEOT using range extent measurements.

Acknowledgments: This work is supported in part by a grant for the National Natural Science Foundation of China NSFC (Nos. U1504619, 61573020, 61671139, 61403414, U1404615 and U1404512), the Scientific Technology Program of Henan Province, China (No. 14210700021), Open Funds of State Key Laboratory of Millimeter Waves (No. K201504), and Natural Science Basic Research Plan in Shaanxi Province of China (2016JQ6070).

Author Contributions: Lifan Sun conceived of the idea and developed the proposed approaches. Jian Lan gave advice on the research and helped edit the paper. Baofeng Ji, Zishu He and Jiexin Pu improved the quality of the manuscript and the completed revision.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Z.; Wu, D.; Xie, W.; Li, L. Tracking the turn maneuvering target using the multi-target Bayes filter with an adaptive estimation of turn rate. *Sensors* **2017**, *17*, 373.
2. Zhang, T.; Wu, R. Affinity propagation clustering of measurements for multiple extended target tracking. *Sensors* **2015**, *15*, 22646–22659.
3. Zhong, Z.; Meng, H.; Zhang, H.; Wang, X. Performance bound for extended target tracking using high resolution sensors. *Sensors* **2010**, *10*, 11618–11632.
4. Li, B.; Mu, C.; Han, S.; Bai, T. Model parameter adaption-based multi-model algorithm for extended object tracking using a random matrix. *Sensors* **2014**, *14*, 7505–7523.
5. Koch, W.; van Keuk, G. Multiple hypothesis track maintenance with possibly unresolved measurements. *IEEE Trans. Aerosp. Electron. Syst.* **1997**, *33*, 883–892.
6. Gilholm, K.; Salmund, D. Spatial distribution model for tracking extended objects. *IEE Proc. Radar Sonar Navig.* **2005**, *152*, 364–371.
7. Mahler, R. PHD filters for nonstandard targets, I: Extended targets. In Proceedings of the 12th International Conference on Information Fusion (Fusion 2009), Seattle, WA, USA, 6–9 July 2009.
8. Lian, F.; Han, C.; Liu, W.; Sun, J. Unified cardinalized probability hypothesis density filters for extended targets and unresolved targets. *Signal Process.* **2012**, *92*, 1729–1744.
9. Zhang, Y.; Ji, H. A novel fast partitioning algorithm for extended target tracking using a Gaussian mixture PHD filter. *Signal Process.* **2013**, *93*, 2975–2985.
10. Vo, B.-N.; Mallick, M.; Bar-Shalom, Y.; Coraluppi, S.; Osborne, R.; Mahler, R.; Vo, B.-T. Multitarget tracking. In *Wiley Encyclopedia of Electrical and Electronics Engineering*; Wiley: Hoboken, NJ, USA, 2015.
11. Koch, W. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Trans. Aerosp. Electron. Syst.* **2008**, *44*, 1042–1059.
12. Feldmann, M.; Franken, D.; Koch, W. Tracking of extended objects and group targets using random matrices. *IEEE Trans. Signal Process.* **2011**, *59*, 1409–1420.
13. Lan, J.; Li, X.R. Tracking of maneuvering non-ellipsoidal extended object or target group using random matrix. *IEEE Trans. Signal Process.* **2014**, *62*, 2450–2463.
14. Lan, J.; Li, X.R. Tracking of extended object or target group using random matrix: New model and approach. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 2973–2989.
15. Baum, M.; Hanebeck, U.D. Extended object tracking with random hypersurface models. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 149–159.
16. Hong, L.; Cui, N.; Pronobis, M.; Scott, S. Local motion feature aided ground moving target tracking with GMTI and HRR measurements. *IEEE Trans. Autom. Control* **2005**, *50*, 127–133.
17. Salmund, D.J.; Parr, M.C. Track maintenance using measurements of target extent. *IEE Proc. Radar Sonar Navig.* **2003**, *150*, 389–395.
18. Klemm, R. Recognition of convoys with array radar. *IET Radar Sonar Navig.* **2012**, *6*, 123–129.
19. Angelova, D.; Mihaylova, L. Extended object tracking using Monte Carlo methods. *IEEE Trans. Signal Process.* **2008**, *56*, 825–832.

20. Ristic, B.; Salmond, D.J. A study of a nonlinear filtering problem for tracking an extended target. In Proceedings of the 7th International Conference on Information Fusion (Fusion 2004), Stockholm, Sweden, 28 June–1 July 2004.
21. Xu, L.; Li, X.R. Hybrid Cramer-Rao lower bound on tracking ground moving extended target. In Proceedings of the 12th International Conference on Information Fusion (Fusion 2009), Seattle, WA, USA, 6–9 July 2009.
22. Sun, L.; Li, X.R.; Lan, J. Modeling of extended objects based on support functions and extended Gaussian images for target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 3021–3035.
23. Sun, L.; Lan, J.; Li, X.R. Modeling for tracking of complex extended object using Minkowski addition. In Proceedings of the 17th International Conference on Information Fusion (Fusion 2014), Salamanca, Spain, 7–10 July 2014.
24. Sun, L.; Lan, J.; Li, X.R. Joint tracking and classification of extended object based on support functions. In Proceedings of the 17th International Conference on Information Fusion (Fusion 2014), Salamanca, Spain, 7–10 July 2014.
25. Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking—Part I: Dynamic models. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1333–1364.
26. Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking—Part V: Multiple-model methods. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1255–1321.
27. Julier, S.J.; Uhlmann, J.K. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control.* **2000**, *45*, 477–482.
28. Lan, J.; Li, X.R. Nonlinear estimation by LMMSE-based estimation with optimized uncorrelated augmentation. *IEEE Trans. Signal Process.* **2015**, *63*, 4270–4283.
29. Lan, J.; Li, X.R. Multiple Conversions of Measurements for Nonlinear Estimation. *IEEE Trans. Signal Process.* **2017**, *65*, 4956–4970.
30. Ito, K.; Xiong, K. Gaussian filters for nonlinear filtering problems. *IEEE Trans. Autom. Control* **2000**, *45*, 910–927.
31. Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269.
32. Baheti, R.; O’Hallaron, D. Efficient parallel implementation of target tracking Kalman filter. In Proceedings of the 27th IEEE Conference on Decision & Control, Austin, TX, USA, 7–9 December 1988; Volume 1, pp. 376–381.
33. Li, X.R.; Zhao, Z.-L. Evaluation of estimation algorithms. Part I: Incomprehensive performance measures. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 1340–1358.
34. Gardner, R.J.; Kiderlen, M.; Milanfar, P. Convergence of algorithms for reconstructing convex bodies and directional measures. *Ann. Stat.* **2006**, *34*, 1331–1374.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

A Clustering-Oriented Closeness Measure Based on Neighborhood Chain and Its Application in the Clustering Ensemble Framework Based on the Fusion of Different Closeness Measures

Shaoyi Liang and Deqiang Han *

MOE KLINNS Lab, Institute of Integrated Automation, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China; shaoyi.liang2@stu.xjtu.edu.cn

* Correspondence: deqhan@mail.xjtu.edu.cn; Tel.: +86-131-191-15975

Received: 21 July 2017; Accepted: 19 September 2017; Published: 28 September 2017

Abstract: Closeness measures are crucial to clustering methods. In most traditional clustering methods, the closeness between data points or clusters is measured by the geometric distance alone. These metrics quantify the closeness only based on the concerned data points' positions in the feature space, and they might cause problems when dealing with clustering tasks having arbitrary clusters shapes and different clusters densities. In this paper, we first propose a novel Closeness Measure between data points based on the Neighborhood Chain (*CMNC*). Instead of using geometric distances alone, *CMNC* measures the closeness between data points by quantifying the difficulty for one data point to reach another through a chain of neighbors. Furthermore, based on *CMNC*, we also propose a clustering ensemble framework that combines *CMNC* and geometric-distance-based closeness measures together in order to utilize both of their advantages. In this framework, the “bad data points” that are hard to cluster correctly are identified; then different closeness measures are applied to different types of data points to get the unified clustering results. With the fusion of different closeness measures, the framework can get not only better clustering results in complicated clustering tasks, but also higher efficiency.

Keywords: clustering; clustering ensemble; closeness measure; geometric distance; neighborhood chain

1. Introduction

Clustering is an important topic in machine learning, which aims to discover similar data and group them into clusters. Various clustering algorithms have been proposed and widely used in different areas such as sensor networks [1–4], image processing [5–7], data mining [8–10], text information processing [11,12], etc.

In classical clustering algorithms, the centroid-based methods, density-based methods and connectivity-based methods are the most commonly used in practice (such a categorization is according to the different cluster models employed). The most well-known clustering methods include the *k*-means [13], DBSCAN [14], CURE (Clustering Using REpresentatives) [15], etc. They respectively belong to the three aforementioned categories. There are also many recent works focused on improving the performance of the classic clustering schemes [16–19], or exploiting novel clustering methods using different closeness measures [20–23].

The effectiveness of clustering methods, to a great extent, is determined by the closeness measure between data points or clusters. In most classical methods, only the geometric distance is used to measure the closeness between data points, and the closeness between clusters is based on the closeness between their representative data points.

However, the distance-based metrics only focus on the geometric closeness. Although these metrics are appropriate for clustering tasks where the data points' distribution conforms to some strong assumptions, e.g., regular shapes and uniform density, they perform not so well in complicated situations. For example, in k -means and its related methods, partitions are formed based on the distance between each data point and each centroid. Such partitioning rule will bring incorrect results when data points belonging to a given cluster are closer to the centroids of other clusters than to the centroid of the given correct cluster [24]. In DBSCAN, the clustering performance depends on the two parameters defining the neighborhood size and density threshold, which are based on some geometric distances [25]. Since the threshold is predefined and fixed, DBSCAN will generate incorrect results, if the densities of the data points in different clusters are varying. In existing agglomerative hierarchical clustering methods [26], sub-clusters are merged according to the closeness measures such as the single linkage and the complete linkage [27], where the closeness is determined by the pairwise geometric distances between inter-cluster representative data points. Due to the similar reasons that cause problems in the aforementioned methods, these agglomerative hierarchical algorithms usually work well only for the spherical-shaped or uniformly-distributed clusters [24].

The limitations of the traditional closeness measures in clustering have attracted much attention, and thus, many approaches using different closeness definitions have been proposed to address the aforementioned problems. One branch of methods uses the clusters' probability distribution information and its closeness definition. For example, Lin and Chen [28] proposed the cohesion-based self-merging algorithm, which measures the closeness between two sub-clusters by computing and accumulating the "joinability" value of each data point in the two clusters. Dhillon et al. [29] used a KL divergence-based [30] clustering scheme to cluster words in the document categorization. KL-divergence is a measure of "distance" between two probability distributions (it is not a true distance metric because it is not symmetric, and it violates the triangle inequality). In the scheme, a word is assigned to a cluster if the distribution of this word has the smallest "distance" (measured by KL divergence) to the weighted sum of the distributions of the words in that cluster. Similar ideas can be seen in [31,32]. Heller et al. [33] and Teh et al. [34] used Bayes rule in a hierarchical clustering scheme to guide the merging process, where each pair of the clusters is assigned a posterior probability based on the Bayesian hypothesis test and the two clusters with the highest posterior probability are merged. By incorporating the distribution information, these methods are more robust to outliers and can deal with data with arbitrary shapes. However, users must know the clusters' probability density functions (pdf) before running the algorithms. Another branch of refined clustering methods are based on the graph theory. Karypis et al. [24] proposed the algorithm of chameleon, where a graph based on k -nearest neighbors is constructed and then cut into sub-clusters. The relative inter-connectivity and the relative closeness are used to determine the closeness between sub-clusters, and the algorithm achieves good results in finding clusters with arbitrary shapes. Similarly, we can see in [35,36], and more recently in [37], that Zhang et al. proposed an agglomerative clustering method where a structural descriptor of clusters on the graph is defined and used as clusters' closeness measure. The properties of graph theory make it very appropriate to describe clustering problems, and the methods based on graph theory perform well in dealing with clusters with arbitrary shapes. However, we find that these methods often fail to adapt to clustering tasks with very different clusters' densities, although they use new closeness measures. In the two branches of refined clustering methods above, different kinds of closeness measures have been proposed to address the problems caused by the geometric distance-based closeness measures. These new measures do not focus on the geometric closeness alone, and they achieve success in many clustering tasks. However, they still have their own limitations. In particular, as mentioned above, these closeness measures either ignore the density information or need strong a priori information.

Therefore, in this paper, we first focus on designing a more comprehensive closeness measure between data points to substitute the traditional geometric distance-based closeness measures in clustering algorithms. The new measure is called the Closeness Measure based on the Neighborhood

Chain (*CMNC*), where the neighborhood chain is a relationship established between two data points through a chain of neighbors. By substituting their original closeness measures with *CMNC*, many simple clustering methods can deal with the complicated clustering tasks with arbitrary clusters shapes and different clusters densities.

Prior to ours, there were some recent works in the literature that also utilized the k -nearest neighbors relationship in clustering problems. For example, Liu et al. [38] proposed a clustering algorithm named ADPC-KNN (Adaptive Density Peak Clustering k NN), where they modified the density peaks' clustering [39] by using the distribution information of k -nearest neighbors of a data point to calculate its local density. Sur et al. [40] proposed a clustering method that forms a cluster by iteratively adding the cluster's nearest neighbor into that cluster (a threshold is defined determining whether this nearest neighbor can be added into the cluster). In a series of work proposed in [41–44], Qiu et al. used an algorithm called nearest neighbor descent and its several modifications to organize the data points into a fully-connected graph "in-tree", and the clustering results can be obtained after removing a small number of redundant edges in the graph. In the nearest neighbor descent algorithm, each data point "descends" to (links to) its nearest neighbor in the descending direction of density. Other similar works utilizing the k -nearest neighbors in clustering can be seen in [45,46].

In the above cited works, the neighborhood relationship is used in many ways and resolves different problems in clustering. However, our work presented in this paper is different from the existing methods. In our work, the neighborhood relationship is used to construct a pair-wise closeness measure between two data points, which incorporates not only the connectivity, but also the density information of data points.

The work in this paper is an extension of our previous preliminary work in [47], where the basic concepts of *CMNC* were preliminarily proposed. In this paper, we provide more detailed definition and analysis about *CMNC*. Furthermore, based on *CMNC*, we also propose a clustering ensemble framework that combines different closeness measures. Due to the involvement of neighborhood relationships, the computational cost of *CMNC* is relatively high. In the proposed framework, we use different closeness measures (*CMNC* and Euclidean distance) for different data points and get the unified clustering results. In this way, we are able to limit the use of *CMNC* to the "least required" number of data points to get the correct clustering results. Therefore, based on the proposed framework, we can get better clustering results and, at the same time, higher efficiency.

The rest of the paper is organized as follows. Section 2 introduces the basics of the traditional clustering methods and their limitations. In Section 3, the neighborhood chain is introduced, and *CMNC* is proposed. The performance of several clustering methods whose closeness measures are substituted with *CMNC* is provided. The clustering ensemble framework based on different closeness measures is proposed and tested in Section 4. Section 5 concludes this paper.

2. Traditional Clustering Methods and Their Limitations

In this section, we briefly recall some representative clustering methods. We focus on their closeness measures and discuss why these measures might cause problems.

2.1. Centroid-Based Clustering Methods

In the centroid-based clustering methods, e.g., the k -means [13] and the k -medoids [48], the closeness between a data point and the existing clusters' centroid (or medoid) determines to which cluster the data point will be assigned. The clusters' centroids are iteratively updated by minimizing the mean square distance of the data points to their assigned cluster's centroids. In such a procedure, the closeness between a data point and a cluster's centroid is measured by the geometric distance alone, which might cause problems.

For example, in Figure 1, two clusters are represented by the hollow and the solid dots. The hollow star and the solid star are the centroids of the two clusters, respectively. In this case, the data points marked with arrows are geometrically closer to the centroid of the other cluster than to the centroid

of their own cluster. Therefore, using such closeness measure, the clustering algorithm will bring incorrect clustering.

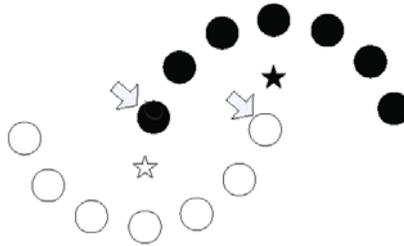


Figure 1. Illustration of problem in the centroid-based clustering method.

2.2. Density-Based Clustering Methods

DBSCAN is a representative density-based clustering method. It needs two predefined parameters *eps* and *minpts*, which respectively determine the closeness threshold and the minimum number of the data points to form a cluster. In DBSCAN, clusters are formed by the data points in the geometrically-dense regions [25]. The data points that are density-connected or density-reachable to each other will be assigned to the same cluster [14]. The parameters *eps* and *minpts* are used to determine whether a group of data is “dense” enough, or can be density-connected to each other. DBSCAN can achieve good performance when dealing with clusters with arbitrary shapes. However, the performance of DBSCAN is very sensitive to the parameters’ selection. Moreover, it may fail when the densities of the clusters are not concordant, even if the parameters are finely selected.

For example, in Figure 2, the hollow, the black solid and the gray solid dots are three different clusters. The densities of the clusters are different: data points in the black cluster are densely distributed, while those in the other two clusters are more sparsely distributed. In such a case, DBSCAN will fail to find all three clusters no matter how the closeness threshold is set. The reason for this problem lies in that the closeness threshold based on the geometric distance is predefined and is not adaptable to the change of clusters’ densities.

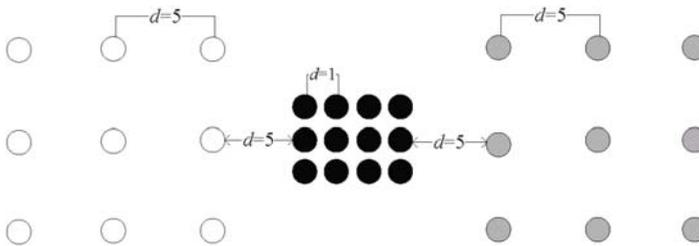


Figure 2. Illustration of the problem in DBSCAN method.

2.3. Connectivity-Based Clustering Methods

Traditional agglomerative hierarchical clustering methods build cluster models based on distance connectivity. In these methods, all the data points are regarded as different clusters in the initial step. They are then iteratively merged according to the closeness measures between clusters. Although there are various methods to quantify the closeness between two clusters [26], these quantifications are mainly based on the geometric distance between two representative data points in the two clusters, and these methods mainly differ in how to choose the representative data points. For example, in the single-link method [49], the closeness between two clusters is represented by the distance between

the geometrically closest pair of data points that respectively belong to the two clusters. CURE [15] is another famous hierarchical algorithm, where the clusters are not represented by a single data point, but a group of well-scattered representative data points. However, its closeness between two clusters is still measured by the geometric distance between the two clusters' representative data points sets, which is essentially the same as other hierarchical methods. Therefore, generally speaking, these agglomerative hierarchical methods all suffer from the problems caused by their geometric distance-based closeness measures. For example, in Figure 3, the true cluster number is two. Assume that in the last iteration of the merging procedure, there are in total three clusters left. Using the aforementioned closeness measures for clusters, the red solid cluster will be merged with the black solid cluster because these two clusters are the closest according to the geometric distance metrics. However, the red cluster is apparently more likely to belong to the same cluster with the hollow cluster from the intuitive point of view.

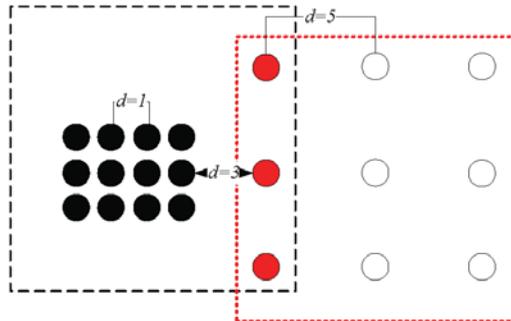


Figure 3. Illustration of the problem in the traditional agglomerative hierarchical methods.

In summary, quantifying the closeness is the key step in clustering algorithms, which determines the assignment of data points and the merging of clusters. The limitations of different categories of clustering methods discussed above are mainly caused by their closeness measures that only emphasize the geometric distances. Therefore, in order to resolve such problems, we propose a more comprehensive measure to quantify the closeness between data points.

3. Measuring Closeness between Data Points Based on the Neighborhood Chain

As discussed in Section 2, using the geometric distance alone to measure the closeness might cause problems. The main reason lies in that under the geometric distance metrics, the closeness between data points is fully determined by the positions of the two points being measured in the feature space, and the influence of any other surrounding data points is ignored. However, in many cases, being geometrically close does not necessarily mean that two data points are more likely to belong to the same cluster.

In this section, we propose a Closeness Measure based on the Neighborhood Chain (CMNC) that quantifies the closeness between data points by measuring the difficulty for one data point attempting to “reach” another through a chain of neighbors. The difficulty is measured by two quantifications called the neighborhood reachability cost and the neighborhood reachability span. Under such a closeness measure, a data point can reach another data point at a low cost as long as they belong to the same cluster, while a data point costs much more to reach another if they belong to different clusters.

Note that the terms “reach” and “reachability” have also appeared in DBSCAN and OPTICS (Ordering Points To Identify the Clustering Structure) [50], describing whether two data points are density connected based on the geometric distance alone. However, in this paper, the “neighborhood reachability” is defined based on the neighborhood relationship between two data points.

3.1. Neighborhood Chain

The neighborhood chain contains a series of data points, including a start point and an end point. Each data point in the chain (except the start point) is one of the k nearest neighbors of its precedent data point. Before giving the formal definition of the neighborhood chain, we first use an example to intuitively illustrate how a chain is established.

Example 1. As shown in Figure 4a, assume that A , B and C are the data points in a dataset. Obviously, B is in the two nearest neighbors of A , and C is in the two nearest neighbors of B . Thus, through an intermediate data point B , a chain from A to C based on the two nearest neighbors is established. We say that C can be reached by A via the chain of two nearest neighbors through B .

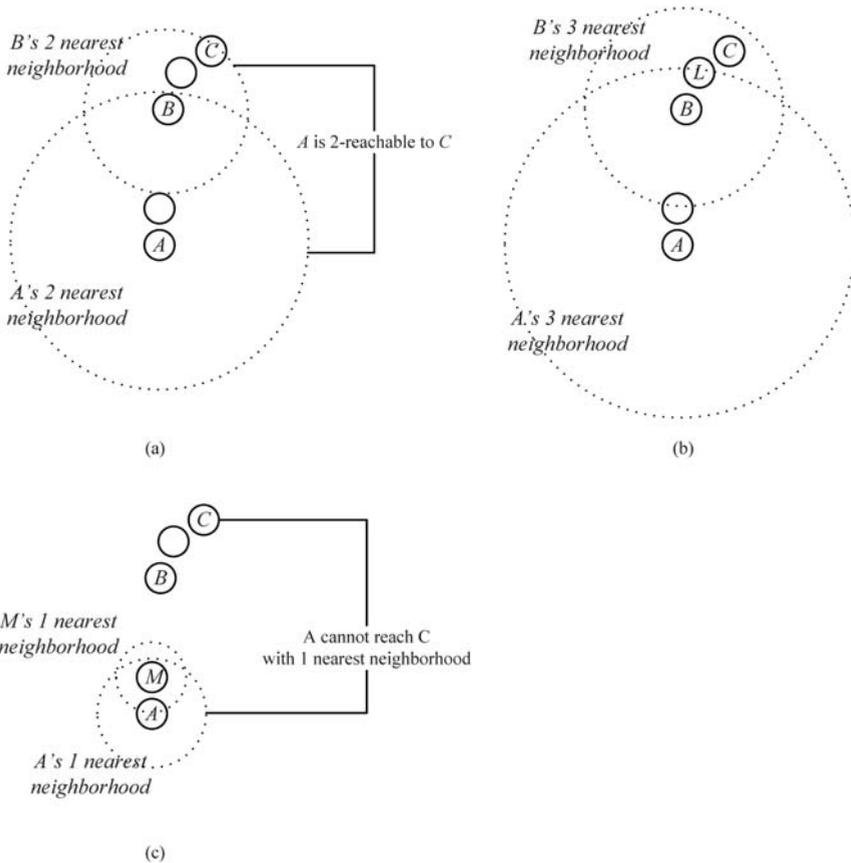


Figure 4. (a) A can reach C based on two nearest neighbors. (b) A can also reach C based on three nearest neighbors. (c) A cannot reach C based on one nearest neighbor.

As shown in Figure 4b, B is also in the three (or more) nearest neighbors of A , and C at the same time is in the corresponding number of the nearest neighbors of B , which means that the chain from A to C based on three (or more) neighbors can also be established. However, the chain established based on the two neighbors takes a lower cost (which means that the required neighbor's number is less).

Actually, in Figure 4, two nearest neighbors comprise the minimum requirement to establish a chain from A to C (Figure 4c shows that the chain from A to C cannot be established based on one nearest neighbor). Therefore, we say that A can reach C through a neighborhood chain with two nearest neighbors, or C is two-reachable from A.

The formal definition of the neighborhood chain is as follows. Assume that $\Omega \subseteq \mathbb{R}^n$ is a dataset and $A, C \in \Omega$. Let $f(A, C, \Omega)$ be a positive integer that makes a set of data points $\{A, M_1, M_2, \dots, M_q, C\}$ in Ω satisfy:

$$\begin{cases} M_1 \in \text{Neighbors}(A, f(A, C, \Omega)) \\ M_i \in \text{Neighbors}(M_{i-1}, f(A, C, \Omega)), 1 < i \leq q \\ C \in \text{Neighbors}(M_q, f(A, C, \Omega)) \end{cases} \quad (1)$$

where $\text{Neighbors}(\cdot, f(A, C, \Omega))$ represents the set containing a data point and its $f(A, C, \Omega)$ nearest neighbors. If such an integer $f(A, C, \Omega)$ exists, we say that the neighborhood chain from A to C can be established.

In the given dataset Ω , $f(A, C, \Omega)$ can take different values to establish different chains from data point A to C. For example, in Example 1 shown above, $f(A, C, \Omega)$ can be 2, 3 or 4 (or even more), which respectively can establish the chain A-B-C, A-L-C, or A-C. Therefore, we define:

$$R(A, C) = \min(f(A, C, \Omega)) \quad (2)$$

as the required neighbor's number to establish the neighborhood chain from A to C. In the rest of this paper, the neighborhood chain refers to the chain established based on the required neighbor's number.

When the required neighbor's number is determined, the corresponding data points $\{M_1, M_2, \dots, M_q\}$ that satisfy Equation (1) are called the intermediate point in the chain from A to C. In Figure 4a, $R(A, C) = 2$, and B is an intermediate point from the data point A to C.

In practice, the required neighbor's number and the corresponding intermediate points can be determined through a "trial and error" process. Such a process can be illustrated by Figure 5.

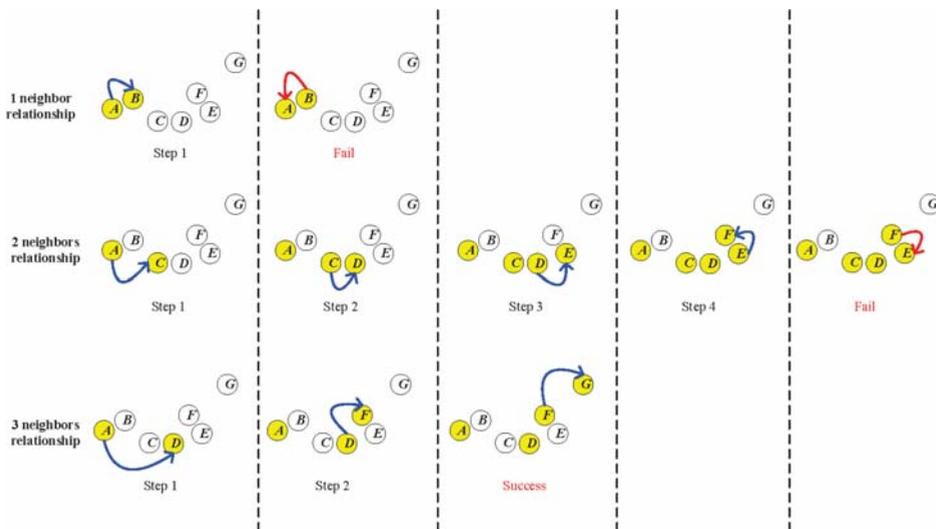


Figure 5. "Trial and error" process to determine the required neighbor's number.

As shown in Figure 5, A, B,...,G are seven data points in a dataset. Now, we would like to establish a neighborhood chain from A to G and determine the required neighbor's number $R(A, G)$. First, we try the one nearest neighborhood relationship. As shown in the first row of Figure 5, in Step 1,

we search the nearest neighbor of the start point A , which is B , and add it into the chain. In Step 2, we continue to search the nearest neighbor of B . However, we find that A is B 's nearest neighbor, and A is already in the chain. Therefore, the searching process enters a loop, and the chain from A to C cannot be established.

Then, we start over to try the two nearest neighbors' relationship. As shown in the second row of Figure 5, in Step 1, we find that B and C are both in the two nearest neighbors of the start point A , and we add C into the chain, because C is closer to the destination (the end point G). In Step 2, we continue to search the two nearest neighbors of C , which is newly added into the chain, and B and D are found. In this step, D is added into the chain because it is closer to G . In Steps 3 and 4, E and F are added into the chain sequentially. However, in Step 5, when we search the two nearest neighbors of the newly added F , we only find E and D , which are both in the chain already. Therefore, the searching process fails in this step, and the chain from A to G cannot be established.

As shown in the third row of Figure 5, we start over to try the three nearest neighbors' relationship. In Step 1, we add D into the chain, because it is the closest to the destination (the end point G) in the three nearest neighbors of A . In Step 2, we find C , E and F in the three nearest neighbors of D , and we add F into the chain for the same reason. In Step 3, we find the end point G in the three nearest neighbors of F , which means that the chain from A to G is successfully established. Therefore, the three nearest neighbors comprise the minimum neighbor's number required to establish the chain, which means that $R(A, G) = 3$.

Along with the determination of $R(A, G)$, the neighborhood chain from A to G is obtained, which is A - D - F - G . $\{D, F\}$ is the set of intermediate points of the chain from A to G .

In practical applications, in the "trial and error" process to establish the neighborhood chain, we might encounter some situations where several data points in the neighborhood of a point (which is newly added into the chain) have the same distance to the end point we want to reach. In such situations, we just randomly choose one of these points to be added into the chain.

3.2. Quantifying the Difficulty to Establish a Neighborhood Chain

In this part, we define two quantifications of the difficulty in establishing a neighborhood chain, which are the neighborhood reachability cost and neighborhood reachability span.

Neighborhood reachability cost (NRC): The neighborhood reachability cost is designed based on the required neighbor's number when establishing a chain from one data point to another. Note that the required neighbor's number is usually not symmetric, i.e., the required neighbor's number from a data point A to another data point C is usually different from that from C to A (as shown in Figure 6). We define a symmetric quantification:

$$NRC(A, C) = f(\max(R(A, C), R(C, A))) \quad (3)$$

as the *neighborhood reachability cost* when establishing a chain between A and C , where $f(\cdot)$ can be a function that is monotonically increasing on $(0, \infty)$. Obviously, the more neighborhood reachability cost needed, the more difficulty in establishing the chain.

In the rest of this paper, $f(\cdot)$ in Equation (3) is designated as the exponential function, because it can make the NRC value grow much faster than the R value grows, which will magnify the difference between the closeness value of two data points from the same cluster and that of two data points from different clusters.

In Equation (3), $\max\{\cdot, \cdot\}$ is used to select the bigger one out of $R(A, C)$ and $R(C, A)$ to make $NRC(A, C)$ a symmetric quantification.

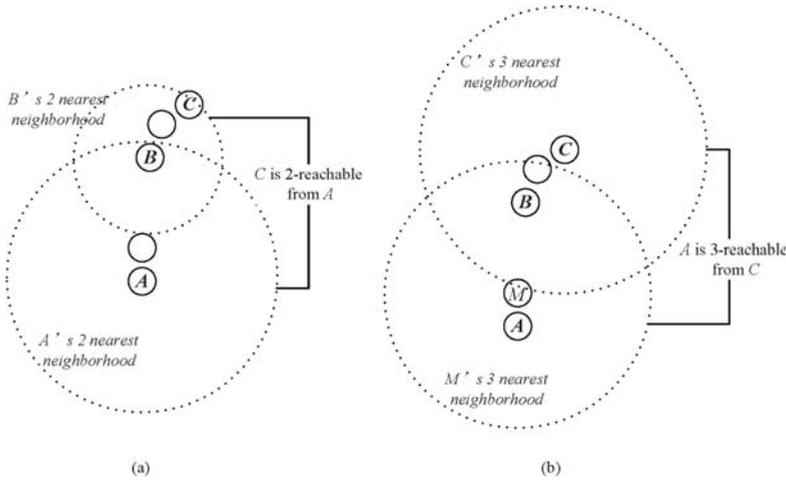


Figure 6. (a) The required neighbor’s number from A to C is 2. (b) The required neighbor’s number from C to A is 3.

Neighborhood reachability span (NRS): Although using the geometric distance alone to measure the closeness between data points might cause problems as previously mentioned, it can still be used as a part of the closeness measure to depict in detail the difficulty in establishing a neighborhood chain. The neighborhood reachability span of a neighborhood chain quantifies the maximum span (distance) between the two adjacent intermediate points in a chain. $\forall A, C \in \Omega$, if $\{M_1, M_2, \dots, M_n\}$ are the intermediate points in the chain from A to C, then there is:

$$S(A, C) = \max\{d(A, M_1), \dots, d(M_{n-1}, M_n), d(M_n, C)\} \tag{4}$$

where $S(A, C)$ is the unidirectional span in the chain from A to C, and $d(\cdot, \cdot)$ is the Euclidean distance between two intermediate points. The *neighborhood reachability span* of the chain between two data points A and C is defined as:

$$NRS(A, C) = \max\{S(A, C), S(C, A)\} \tag{5}$$

By selecting the bigger one out of the two unidirectional spans, the NRS is also a symmetric quantification.

3.3. Closeness Measure between Data Points Based on the Neighborhood Chain

The neighborhood reachability cost and the neighborhood reachability span are two parts that jointly quantify the difficulty to establish a neighborhood chain between two data points, and the difficulty in establishing the chain can be used to measure the data points’ closeness. The CMNC between any two data points A and C in a dataset is defined as:

$$CMNC(A, C) = \frac{1}{NRC(A, C) \cdot NRS(A, C)} \tag{6}$$

A bigger CMNC value means that the chain between the two data points can be more easily established, which represents that the two points are closer, while a smaller CMNC represents the opposite. Strictly speaking, CMNC is not a distance metric since it violates the triangle inequality due to the use of the neighborhood relationship. However, using CMNC as a kind of closeness (similarity) measure, we can obtain more intuitive and rational closeness quantifications compared

with using traditional closeness metrics based on the geometric distance alone in clustering tasks. The followings are two examples illustrating the computation of CMNC.

Example 2. As shown in Figure 7a,b, assume that the distance between any two adjacent data points belonging to the same cluster is one (e.g., the distance between M_1 and M_2 is one) and the distance between the nearest pair of data points that belong to different clusters is 1.5 (i.e., the distance between A and C is 1.5).

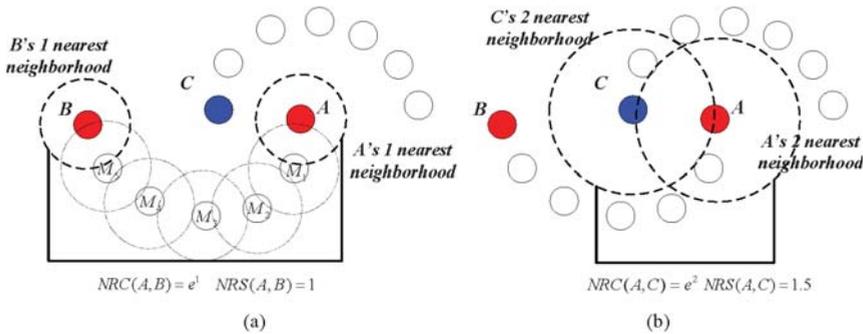


Figure 7. (a) Value of the closeness Measure based on the Neighborhood Chain (CMNC) between A and B. (b) Value of CMNC between A and C.

In Figure 7a, we calculate the CMNC value between data points A and B. Note that M_1 is in the nearest neighborhood of A; M_2 is in the nearest neighborhood of M_1 ; and the relationship spread all the way to B. Therefore, $R(A, B) = R(B, A) = 1$, and we have $NRC(A, B) = e^{\max\{RC(A,B), RC(B,A)\}} = e^1 \approx 2.72$. M_1 to M_5 are the intermediate points in the chain between A and B, and the distance between any two adjacent intermediate points is one. Therefore, we have $NRS(A, B) = \max\{S(A, B), S(B, A)\} = 1$. Then, $CMNC(A, B) = 1 / [NRC(A, B) \cdot NRS(A, B)] = 0.37$.

In Figure 7b, we calculate the CMNC value between data points A and C. Note that two nearest neighbors are needed to establish a chain between A and C, so we have $NRC(A, C) = e^2 \approx 7.40$ and $NRS(A, C) = 1.5$. Therefore, $CMNC(A, C) = 1 / [NRC(A, C) \cdot NRS(A, C)] = 0.09$.

In this example, we see that although the geometric distance between A and B is much longer than that between A and C, A and B is much “closer” than A and C using the CMNC measure.

Example 3. As shown in Figure 8a, the chain between A and B can be established with one nearest neighbor, thus $NRC(A, B) = e^1$ and $NRS(A, B) = 5$. In Figure 8b, C can be reached by A with one nearest neighbor, and thus, $R(A, C) = 1$. However, the lowest cost that it takes for A to be reached by C is 12 nearest neighbors, which means that $R(C, A) = 12$. Therefore, we have $NRC(A, C) = e^{12}$. This shows that although the geometric distance between A and C is equal to that between A and B (e.g., the Euclidean distances between A and C and that between A and B are both five in this case), the data points from two different clusters can be clearly separated using the CMNC measure ($CMNC(A, C) = 1 / (5e^{12}) \approx 1.2 \times 10^{-6}$ is much smaller than $CMNC(A, B) = 1 / (5e^1) \approx 7.4 \times 10^{-2}$).

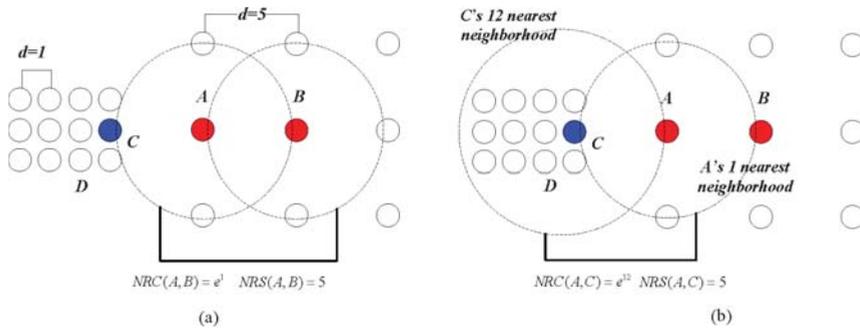


Figure 8. (a) CMNC value between A and B in clusters with different scales. (b) CMNC value between A and C in clusters with different scales.

In fact, in the case shown in Figure 8, the neighborhood reachability cost between any two data points that belong to the same cluster is always e^1 (e.g., $NRC(A,B) = NRC(C,D) = e^1$). This means that although the densities of the two clusters are different from the global point of view, the relationship between data points within each cluster, from the clusters’ local point of view, are very similar. In other words, this means that seeing from the individual cluster they belong to, the closeness between A and B is similar to that between C and D. In such a situation, the closeness of data points depicted by CMNC can adapt to different clusters’ “local density”, while traditional density-based methods like DBSCAN using geometric distance metrics alone can only deal with the clusters having the same density.

In the last part of this section, we give the algorithm to compute the CMNC value in pseudocode. The Algorithm 1 is as follows:

Algorithm 1

Input: $Start_Point = A, End_Point = C, k = 1, Result_Set = \{A\}$, and the dataset Ω .

Output: the value of $CMNC(A,C)$.

- S1: find $Start_Point$'s k nearest neighbors in Ω , and denote them as kNN_set .
 - S2: In kNN_set , find the point closest to End_Point , and denote it as $temp_Point$. If multiple points in kNN_set have the same distance to End_Point , choose one randomly, and denote it as $temp_Point$.
 - S3: If $temp_Point == End_Point$, GO TO S5. Otherwise, GO TO S4.
 - S4: If $temp_Point$ is found in $Result_Set$, then set $Result_Set = \{A\}, Start_Point = A, k = k + 1$, and GO TO S1.
 - Otherwise, $Result_Set = Result_Set \cup temp_Point, Start_Point = temp_Point$, and GO TO S1.
 - S5: If this is the first time entering S5, then $R(A,C) = k$, and $S(A,C)$ equals the maximum distance between any two adjacent points in $Result_Set$. Set $k = 1, Result_Set = \{C\}, Start_Point = C, End_Point = A$, and GO TO S1.
 - Otherwise, $R(C,A) = k$, and $S(C,A)$ equals the maximum distance between any two adjacent points in $Result_Set$. GO TO S6.
 - S6: Calculate $NRC(A,C) = e^{\max\{R(A,C),R(C,A)\}}$,
 $NRS(A,C) = \max\{S(A,C),S(C,A)\}$, and $CMNC(A,C) = 1/NRC(A,C) \cdot NRS(A,C)$.
-

3.4. Computational Complexity of CMNC

In the worst case, the computational cost to calculate $CMNC(A,C)$ could be $O(n(n-1)^2)$. In the following, we will illustrate how the expression is obtained.

In order to quantify the computational cost when measuring the closeness between two data points (say A and C) with CMNC, we need to examine the computational cost of each component (NRC and NRS) of the CMNC measure.

Computational cost to calculate $NRC(A,C)$: As shown in Equation (3), we need first to compute $R(A,C)$ and $R(C,A)$ before computing $NRC(A,C)$. As illustrated in Section 3.1, we use a “trial and

error” process to build the neighborhood chain from A to C . Assume that we need to try t times to successfully build the chain from data point A to C (which means that the chain is established on the t -th try, and $R(A, C) = t$), and in each try i , $0 < i < t$, we have added m_i points into the (unaccomplished) chain before the establishing of the chain fails. In the t -th try, the chain is established; therefore, m_t is the number of the intermediate points from A to C . Under such assumptions, we actually need to execute $\sum_{i=1}^t m_i$ times the nearest neighbor searching algorithm, where the distance between one data point (whose neighbors are to be found) and all other data points in the dataset will be computed. Therefore, the computational cost calculating $R(A, C)$ can be expressed as $O(n \cdot \sum_{i=1}^t m_i)$, where n is the number of data points. Similarly, we assume that $R(C, A) = t^*$ (which means that the chain from C to A is established on the t^* -th try), and in each try to establish the chain from C to A , m_j^* ($0 < j \leq t^*$) represents the number of data points added into the chain. The computational cost calculating $R(C, A)$ can be expressed as $O(n \cdot \sum_{j=1}^{t^*} m_j^*)$. Therefore, the computational cost calculating $NRC(A, C)$ can be obtained by summing the cost of $R(A, C)$ and $R(C, A)$ and be expressed as $O\left(n \cdot \sum_{0 < i \leq t, 0 < j \leq t^*} (m_i + m_j^*)\right)$.

Computational cost to calculate $NRS(A, C)$: As shown in Equations (4) and (5), we need to compute the distance between each pair of the adjacent intermediate points in the chain from A to C and that from C to A . Therefore, under the assumptions in the previous paragraph, the computational cost calculating $NRS(A, C)$ can be expressed as $O(m_t + m_{t^*}^*)$.

The total computational cost to calculate $CMNC(A, C)$ can be obtained by summing the cost of $NRC(A, C)$ and $NRS(A, C)$. In normal situations, m_t and $m_{t^*}^*$ are much less than the data points number n ; therefore, the cost of $NRS(A, C)$ is negligible. The computational cost to calculate $CMNC(A, C)$ can be expressed as $O\left(n \cdot \sum_{0 < i \leq t, 0 < j \leq t^*} (m_i + m_j^*)\right)$.

The expression illustrates that the computational cost to calculate $CMNC(A, C)$ is determined by the total execution times of the nearest neighbor searching algorithm in the establishing of the neighborhood chain from A to C and from C to A . Such an expression of the computational cost depends largely on the inner structures of the given dataset, and the selected start and end points; therefore, we can hardly use it to evaluate the average computational cost to calculate the $CMNC$ value between any two data points in a dataset. However, we can still use it to estimate the highest possible cost to calculate $CMNC(A, C)$ in the worst case. In the extreme situation, t and t^* can both reach $n - 1$, and m_i and $m_{j^*}^*$ can also reach $n - 1$. Therefore, in the worst case, the computational cost to calculate $CMNC(A, C)$ could be $O\left(n(n - 1)^2\right)$.

3.5. Substituting Closeness Measures in Traditional Clustering Methods with $CMNC$

In this part, we test the performance of the proposed closeness measure by substituting the geometric-distance-based closeness measures between data points in some clustering methods with $CMNC$. The methods for testing include the original versions of k -medoids, single-link, CURE and their $CMNC$ -based versions (the closeness between the data points or between clusters in these methods is calculated with pairwise-distance-based measures; thus, it is easy to substitute these measures with $CMNC$). DBSCAN is also tested for comparison.

In the test of all the original version methods, Euclidean distance is used as the closeness measure. In the test of k -medoids and $CMNC$ -based k -medoids, to exclude the impact of initial centers selection, we give them the same initial centers, where each center is randomly selected from one natural cluster. In $CMNC$ -based k -medoids, the “distance” between data points and the centers is calculated with $CMNC$. In $CMNC$ -based single-link method, the similarity of two clusters is represented by the

CMNC value of their most similar (using CMNC measure) pair of data points. In CMNC-based CURE method, the clusters with the closest (using CMNC measure) pair of representative data points are merged in each iteration. The natural (true) clusters number is assigned to k -medoids, single-link, CURE and their CMNC-based versions as the input parameter.

The test results of the aforementioned methods on three datasets [51–53] are shown in Figures 9–11. For DBSCAN, the shown results are the best results selected by traversing its parameters eps and $minpts$.

In Figure 9, there are two natural clusters, and the two clusters have different densities and twisted shapes. In Figure 10, four natural clusters can be found, in which two clusters have the shapes of concentric circles and the other two have very different densities. In Figure 11, there are three clusters in total, and two of them are spherically distributed. However, the third one that surrounds the two clusters makes the situation more complicated.

These datasets are difficult for the test using traditional clustering methods, and they fail to find all the clusters correctly as shown in Figures 9–11. However, by substituting the closeness measures, the CMNC-based methods successfully find the correct clusters in all the tested datasets. In this test, DBSCAN can also handle the non-spherical clusters shapes. However, it cannot deal with the clusters having different densities. For example, in Figure 10, if a big eps is set, the two natural clusters in the upper-right corner will be considered as a whole cluster, while if a small eps is set, the data points in the sparsely-distributed cluster will all be considered as the noise.

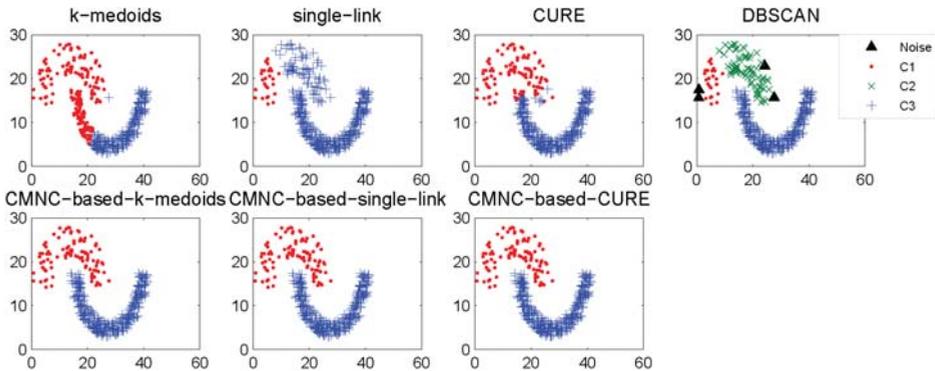


Figure 9. Comparison of clustering results on the “double-moon” dataset.

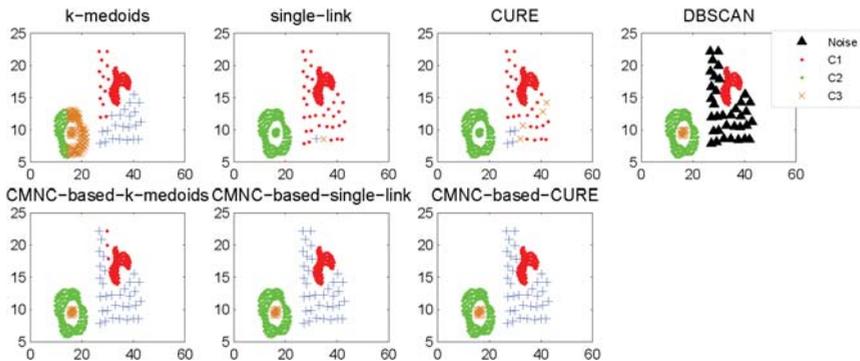


Figure 10. Comparison of clustering results on the “compound” dataset.

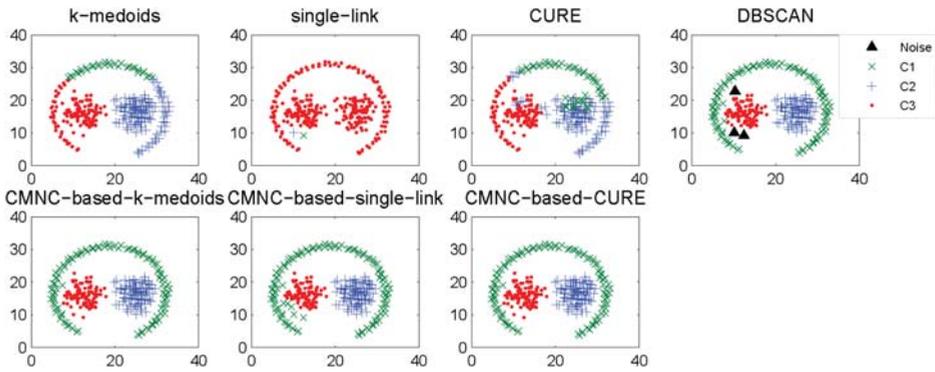


Figure 11. Comparison of clustering results on the “path-based” dataset.

We have also tested the clustering methods on some UCI datasets [54]. The Normalized Mutual Information (NMI) [55] and the Clustering Error (CE) [56] are used as the quantitative criterion for the performance evaluation of the tested methods.

NMI provides an indication of the shared information between a pair of clusters [55]. The bigger this NMI value, the better the clustering performance. For CE, obviously, a lower value is preferred. The test results are shown in Tables 1 and 2. For *k*-medoids and *CMNC*-based *k*-medoids, the results shown are the average results of 20 runs. For *CURE* and *CMNC*-based *CURE*, the results shown are the best results found by traversing the needed parameters.

Table 1. Clustering Error (CE) results of the tested methods.

	<i>k</i> -Medoids	<i>CMNC</i> -Based <i>k</i> -Medoids	Single-Link	<i>CMNC</i> -Based Single-Link	<i>CURE</i>	<i>CMNC</i> -Based <i>CURE</i>
<i>iris</i>	16.67%	14.67%	32.67%	32.00%	32.67%	30.67%
<i>glass</i>	43.27%	40.19%	62.62%	61.68%	59.81%	52.34%
<i>wine</i>	30.11%	28.65%	56.74%	41.57%	35.39%	31.46%
<i>DataUser Modeling</i>	47.09%	36.43%	63.57%	63.18%	64.73%	51.16%

Table 2. Normalized Mutual Information (NMI) results of the tested methods.

	<i>k</i> -Medoids	<i>CMNC</i> -Based <i>k</i> -Medoids	Single-Link	<i>CMNC</i> -Based Single-Link	<i>CURE</i>	<i>CMNC</i> -Based <i>CURE</i>
<i>iris</i>	0.6957	0.7151	0.7355	0.7452	0.7229	0.7452
<i>glass</i>	0.3148	0.3668	0.1196	0.1415	0.1751	0.3523
<i>wine</i>	0.4199	0.4326	0.0914	0.3041	0.3982	0.4392
<i>DataUser Modeling</i>	0.2316	0.3173	0.0774	0.0822	0.0490	0.1098

Note that the *CE* and *NMI* results of *k*-medoids and *CMNC*-based *k*-medoids methods are obtained by averaging the results of 20 runs, where their initial centers are chosen randomly, so we need further to implement a test of significance to validate that the results of *CMNC*-based *k*-medoids are significantly better than that of the original *k*-medoids method. The methodology we use in this paper is the *t*-test [57]. In the test, we assume that the *CE* and *NMI* results of each run of *k*-medoids and *CMNC*-based *k*-medoids come from two normal distributions that have the same variance. The null

hypothesis (H_0) is that the mean of the *CE* (or *NMI*) results of *CMNC*-based k -medoids equals that of the original k -medoids method. On the contrary, H_1 represents that the mean values of the two groups of data are statistically different (under certain significance level). If H_0 holds, we have:

$$\frac{\bar{X} - \bar{Y}}{S_w \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t(n_1 + n_2 - 2) \tag{7}$$

where \bar{X} and \bar{Y} respectively represent the mean of the *CE* (or *NMI*) results obtained by *CMNC*-based k -medoids and the original k -medoids methods, n_1 and n_2 respectively represent the number of instances in X and Y and:

$$S_w = \sqrt{\frac{(n_1 - 1)S_{1n_1}^2 + (n_2 - 1)S_{2n_2}^2}{n_1 + n_2 - 2}} \tag{8}$$

where $S_{1n_1}^2$ and $S_{2n_2}^2$ respectively represent the variance (using Bessel’s correction) of the two sets of results.

If the observation of the t -statistic $t = \frac{\bar{X} - \bar{Y}}{S_w \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$ falls into the rejection region, which means that $|t| \geq t_{\alpha/2}(n_1 + n_2 - 2)$, then H_0 will be rejected, representing that the mean values of the *CE* and *NMI* results obtained by *CMNC*-based k -medoids are statistically different from those obtained by the original k -medoids method. The test results are shown in Tables 3 and 4.

Table 3. Significance test of *CE* results: *CMNC*-based k -medoids vs. k -medoids.

	<i>t</i> -Statistic Value	Rejection Region ($\alpha = 0.1$)	$H_0: \text{mean}(CE_{CMNC\text{-based-}k\text{-medoids}}) = \text{mean}(CE_{k\text{-medoids}})$
<i>iris</i>	-2.5131	$ t > 1.6860$	rejected
<i>glass</i>	-3.0482	$ t > 1.6860$	rejected
<i>wine</i>	-1.8074	$ t > 1.6860$	rejected
<i>DataUser Modeling</i>	-2.9970	$ t > 1.6860$	rejected

Table 4. Significance test of *NMI* results: *CMNC*-based k -medoids vs. k -medoids.

	<i>t</i> -Statistic Value	Rejection Region ($\alpha = 0.1$)	$H_0: \text{mean}(NMI_{CMNC\text{-based-}k\text{-medoids}}) = \text{mean}(NMI_{k\text{-medoids}})$
<i>iris</i>	2.8505	$ t > 1.6860$	rejected
<i>glass</i>	7.5803	$ t > 1.6860$	rejected
<i>wine</i>	3.7374	$ t > 1.6860$	rejected
<i>DataUser Modeling</i>	3.6979	$ t > 1.6860$	rejected

The results in Tables 3 and 4 illustrate that we have sufficient reasons to reject the null hypothesis H_0 on all the tested datasets (under significance level $\alpha = 0.1$), which means that the results obtained by *CMNC*-based k -medoids are statistically better (lower under *CE* index and higher under *NMI* index) than those of the original k -medoids method.

Synthesizing the clustering results shown in Figures 9–11 and the test results in Tables 1–4, it can be concluded that, on the tested datasets, by substituting the closeness measures with *CMNC*, the *CMNC*-based methods can counter-act the drawbacks of the traditional methods and generate correct results in clustering tasks with arbitrary cluster shapes and different cluster scales. The results also show that the *CMNC*-based methods can work well for the tested real-world data and can achieve better performance (under *CE* and *NMI* indexes) than their original versions.

4. Multi-Layer Clustering Ensemble Framework Based on Different Closeness Measures

In previous sections, we proposed *CMNC* to deal with the problems brought by the closeness measures based on geometric distance and achieved good results in clustering tasks with arbitrary cluster shapes and different cluster densities. However, as shown in Section 3.4, the computational cost of *CMNC* is high due to the involvement of neighborhood relationships. Actually, in many simple clustering tasks, or for the “simple part” of some complicated tasks, the geometric distance-based closeness measures can also lead to satisfactory clustering results. They can handle these simple tasks with low computational cost, and they are also easy to implement. Therefore, in this section, we try to incorporate the advantages of *CMNC* and geometric distance-based closeness measures to deal with complicated clustering tasks with higher clustering accuracy and, at the same time, higher time efficiency.

In order to combine the two kinds of closeness measures, we propose a multi-layer clustering ensemble framework. In this framework, the data points that are hard to group into the correct cluster (we call them the “bad data points”, e.g., the data points in the overlapping regions of two non-spherical shaped clusters) can be identified. Thus (in prototype-based clustering schemes), we can apply *CMNC* only to these data points when calculating their closeness to the clusters’ centroids. In this way, the new framework can retain the low computational cost in simple clustering tasks, where only a few “bad data points” need to be dealt with; while in complicated cases, the new framework can achieve much better clustering accuracy than traditional ones and not much computational cost due to the selective application of *CMNC*.

In the framework, a group of *k*-medoids algorithms with random *k* values ($2 \leq k \leq k_{\max}$) and *k* initial centroids (in the rest of this paper, we will call them the “member clusterers”) run repeatedly from Layer-1 to Layer-*T*. In the first layer, Euclidean distance is used as the closeness measure of all pairs of data points. In the following layers, along with the execution of the “bad data points” identification algorithm on the previous layer’s output, some data points will be identified to be the “bad data points”, and *CMNC* will be used when calculating the “distance” between these “bad data points” and the clusters’ centroids in the member clusterers. The identification algorithm will be executed once in each new layer based on the outputs of the previous layer, and the iteration ends when no more “bad data points” (compared with the “bad data points” number found in the previous layer) are found in a certain layer, or when the user assigned maximum layer’s iteration number *T* is met. One additional *k*-medoids (or other clustering methods) will be executed based on the finally found “bad data points” and normal data points to generate the final clustering results. Figure 12 shows an illustration of the proposed framework.

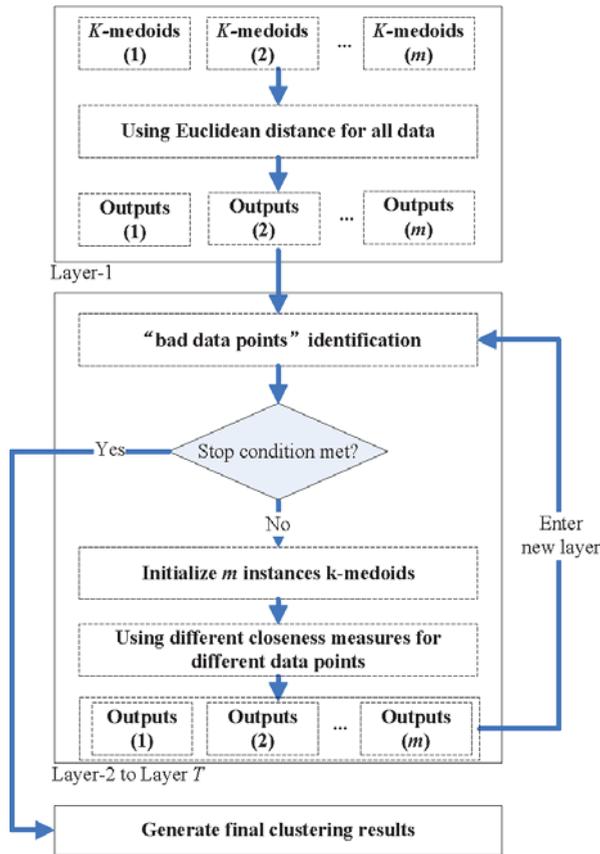


Figure 12. Illustration of the implementation of the proposed framework.

4.1. Output of One Layer

In the proposed framework, a group of member clusterers runs in each layer. Assume n is the data point’s number in a dataset. Each clusterer will generate a $n \times n$ matrix called the Partitioning Matrix (PM), where $PM(i, j) = 1$ if the i -th and the j -th data points belong to the same cluster, or zero if otherwise. Assume m is the clusterer’s number. In each layer, we can obtain m instances of PM.

By extracting the value in position (i, j) from each PM, we can obtain a vector with the length m indicating the m clusterers’ judgments on whether the i -th and the j -th data points belong to the same cluster. We call this vector the Judgment Vector (JV).

Each pair of data points can generate one judgment vector, therefore, the outputs of one layer will be $n(n - 1)/2$ instances of JV (the vectors generated for (i, j) and (j, i) are the same, and any data point must be in the same cluster with itself).

4.2. Identification of “Bad Data Points”

The “bad data points” refers to those data points that cannot be easily clustered correctly, e.g., the data points in the overlapping regions of two clusters. One possible pair of “bad data points” can be characterized by one JV that has elements with high discrepancy. It indicates that the clusterers have very different judgments on whether this pair of data points belongs to the same cluster, which means that these two data points might be hard to cluster correctly under the available conditions.

The discrepancy of the elements in one JV can be quantified by Shannon entropy as:

$$H(JV) = -\left[\frac{m-\Delta}{m} \log_2\left(\frac{m-\Delta}{m}\right) + \frac{\Delta}{m} \log_2\left(\frac{\Delta}{m}\right)\right] \quad (9)$$

where Δ is the number of “1” elements in JV.

By calculating the entropy of each JV output from the previous layer, we can obtain an $n \times n$ matrix where the element in position (i, j) indicates the quantified discrepancy of the clusterers’ judgments on data points pair (i, j) . We call this matrix the Entropy Matrix (EM). Assume W is a data point. In the dataset, there are n different pairs of data points that contains W . In these n pairs of data points, if the number of the pairs that have relatively high discrepancy is relatively large, then there will be a higher possibility that W is a “bad data point”.

Therefore, we define data point W to be a “bad data point” if:

$$\text{Card}[\text{high_En}(W, \alpha)] \geq \beta n, \alpha \in (0, 1), \beta \in (0, 1) \quad (10)$$

where $\text{Card}[]$ returns the elements’ number in a set, β is a user-defined parameter and $\text{high_E}(W, \alpha)$ finds the elements in the w -th row (w is the order number of data point W in the n data points) of EM whose values are bigger than another user-defined parameter α .

In Equation (10), α and β are two parameters influencing the number of “bad data points” identified in a certain layer. The larger α and β are, the less “bad data points” will be found.

4.3. Output of the Clustering Ensemble Framework

After the “bad data points” are found in the previous layer, the clusterers in a new layer can generate new PMs based on different closeness measures. In this new layer, the clusterers will apply the CMNC metric when calculating the “distance” between the clusters’ centroids and the “bad data points”.

After obtaining the new PMs, we can further obtain the new JVs and the new EM. The “bad data points” identification algorithm will run again in this new layer and output the identified “bad data points” to the next new layer. This iteration stops when no more “bad data points” are found in a certain layer, or the user-given maximum iteration number T is met.

One additional instance of clustering methods will be executed using different closeness measures based on the finally found “bad data points” and normal data points to generate the final clustering results.

Following, we give an outline of the proposed clustering ensemble framework (See Algorithm 2).

Algorithm 2

Input: n data points, number of clusterers m , max clusters number k_{max} , max iteration number T , parameters α, β .

Output: data partitioning

Steps:

- S1. Initialize m member clusterers (k -medoids) with random k values (constrained to k_{max}) and random k clusters centroids.
 - S2. Calculate PM_1 to PM_m with the m clusterers. If this is not the first time entering S2, then CMNC is applied to the “bad data points”. Otherwise, only Euclidean distance is used.
 - S3. Extract JV for every pair of data points from PM_1 to PM_m .
 - S4. Calculate information entropy on each JV, and generate EM.
 - S5. Identify “bad data points” based on EM.
 - S6. If no more “bad data points” are identified compared with the last iteration, or the iteration number reaches T , GO TO S7. Else, enter a new layer and GO TO S1.
 - S7. Generate the partitioning on the dataset. Return the clusters found.
-

4.4. Experiments

In this section, we will test the proposed clustering ensemble framework.

4.4.1. Influence of Parameter Selection

In the proposed framework, parameters that need to be assigned by the user include the number of member clusters m , the number of max clusters k_{max} , number of max iterations T and the parameters α, β .

The settings of m, k_{max} and T do not significantly influence the clustering results, while α and β are two major parameters that influence the identification of “bad data points”. In the following, we first examine how the two parameters can influence the “bad data points” detection. The test is run on the “double-moon” dataset, which contains two clusters and 400 data points.

Figure 13 shows the number of the “bad data points” found by the identification algorithm under different α and β combinations. Figures 14–16 locate the “bad data points” (the blue asterisks) found under three certain parameter combinations. The figures illustrate that along with the increase of the number of “bad data points” found, the remaining data points (the dots) become much more easier to cluster correctly with simple clustering algorithms.

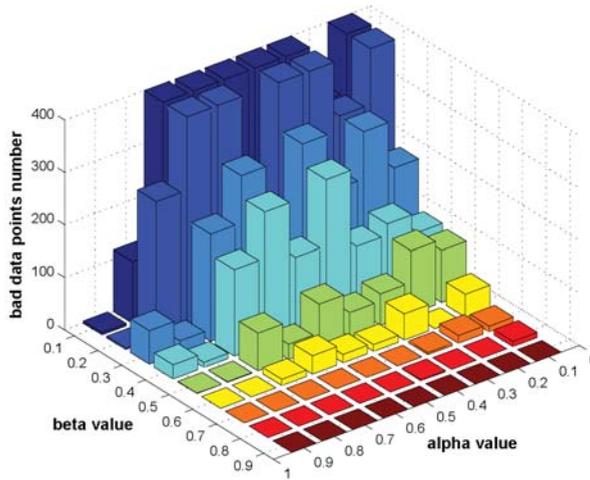


Figure 13. The influence of α and β on the “bad data points” identification, $m = 5$.

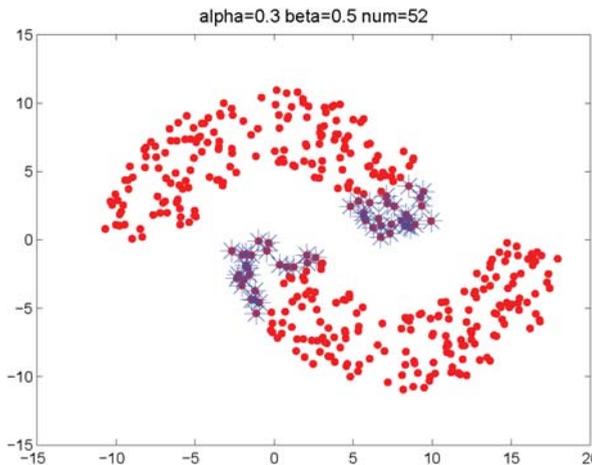


Figure 14. The “bad data points” found under $\alpha = 0.3, \beta = 0.5$ and $m = 5$.

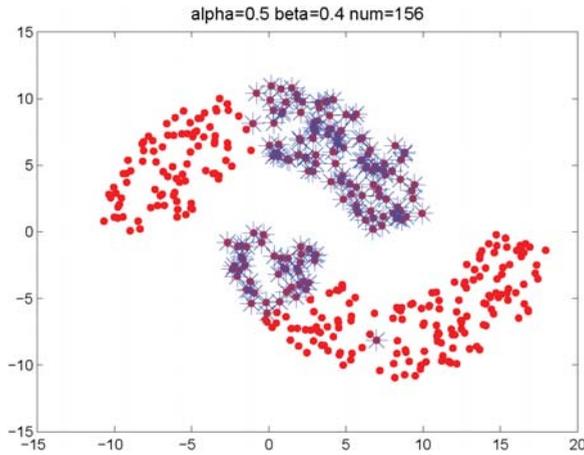


Figure 15. The “bad data points” found under $\alpha = 0.5$, $\beta = 0.4$ and $m = 5$.

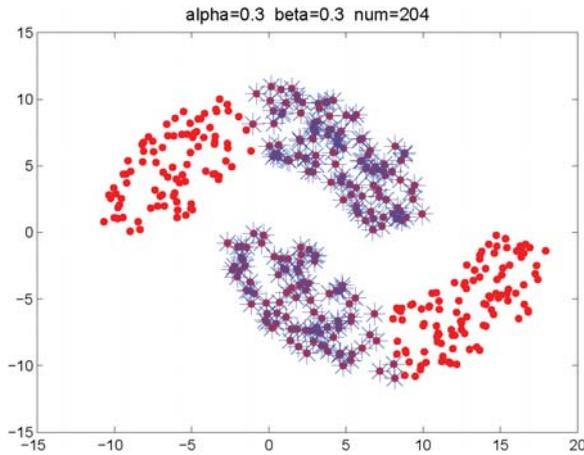


Figure 16. The “bad data points” found under $\alpha = 0.3$, $\beta = 0.3$, and $m = 5$.

In order to more generally examine the effect of parameters α and β , we further test the parameter combinations on several more datasets. We first change the data point’s number in the “double-moon” dataset, varying from 200–1000, which is shown in Figure 17. The numbers of the “bad data points” found in these datasets under different α and β combinations are shown in Figure 18.

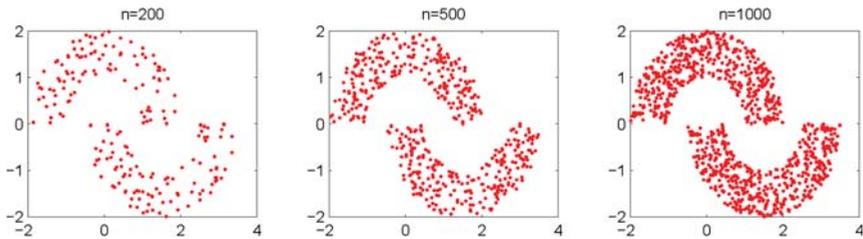


Figure 17. Synthetic dataset with different data numbers from 200 to 1000.

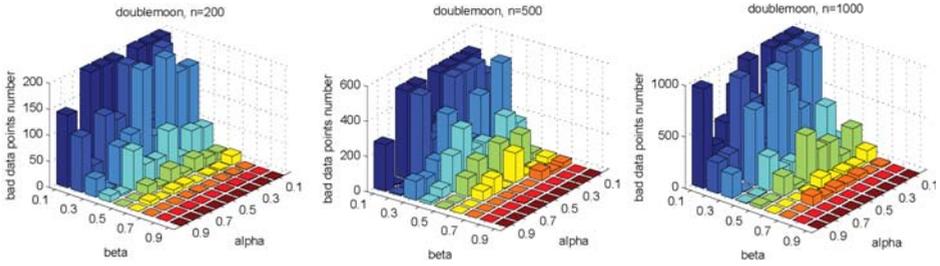


Figure 18. Number of “bad data points” found under different α and β combinations in datasets with different data numbers.

We also test the parameter combinations on another synthetic dataset “path-based” (the dataset is shown in Figure 11), and the numbers of the “bad data points” found under different parameter combinations are shown in Figure 19.

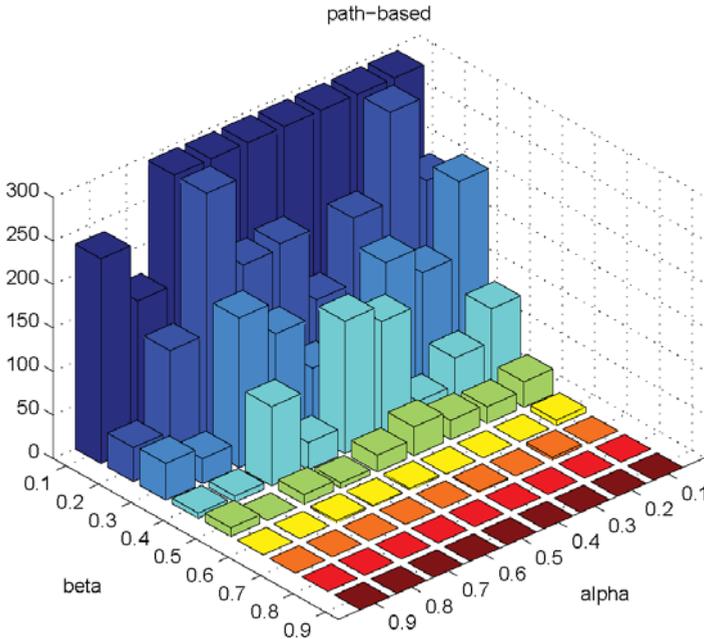


Figure 19. Number of “bad data points” found under different α and β combinations in the “path-based” dataset.

The results in Figures 18 and 19 show that, in different datasets, parameters α and β influence the number of the “bad data points” in similar ways. Therefore, the changing tendency of the “bad data points” number presented under different parameter combinations illustrated in the figures can be seen as a general guide for the tested datasets when choosing the parameters in practice.

4.4.2. Clustering Results of the Proposed Framework

In this section, we give the clustering results of the proposed clustering ensemble framework on some 2D datasets. The corresponding parameter settings are provided for each case.

As shown in Figures 20–22, all the tested datasets have non-spherically-shaped clusters. These datasets usually cannot be correctly clustered with the simple centroid-based clustering methods, or the clustering ensemble methods based on them. However, by applying the *CMNC* metric to part of the data points in the datasets, our proposed clustering ensemble framework can generate very good results on these datasets.

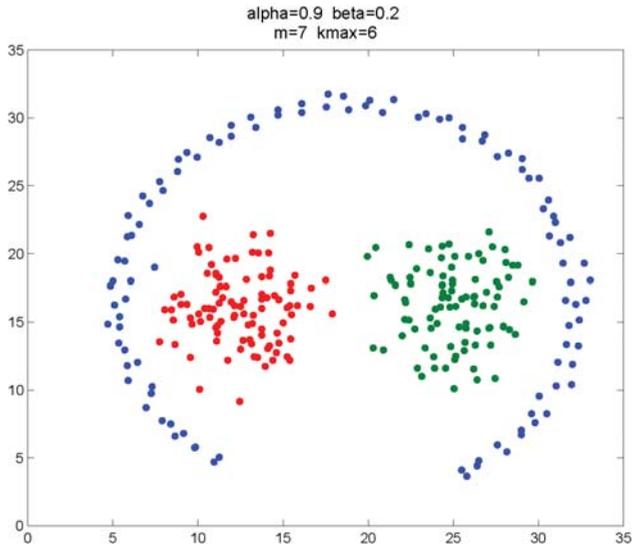


Figure 20. Clustering result on the "path-based" dataset.

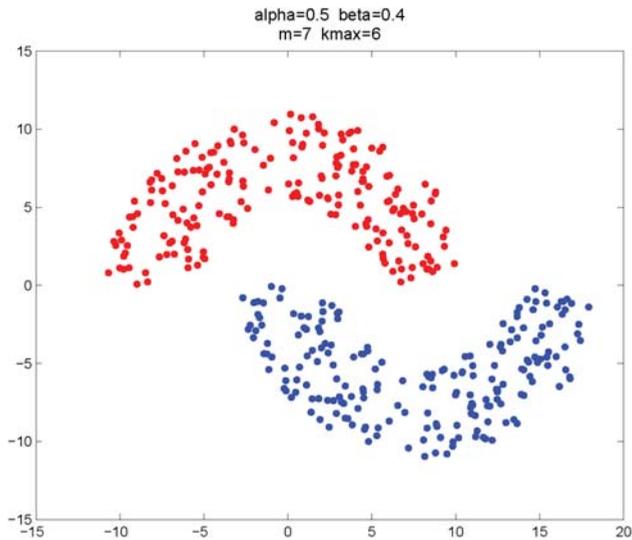


Figure 21. Clustering result on "double-moon" dataset.

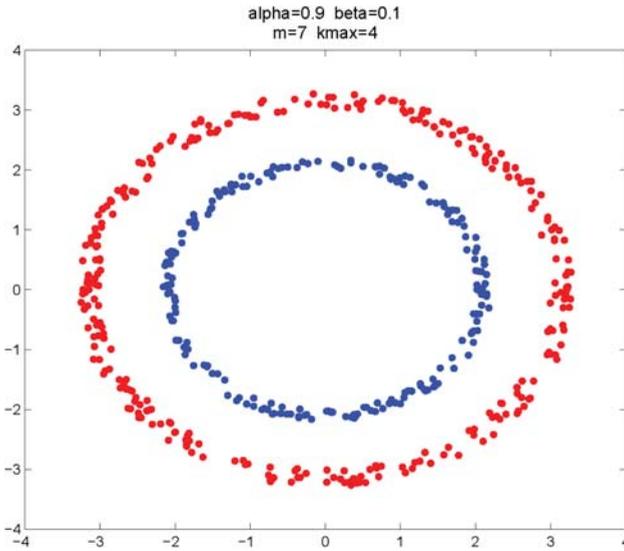


Figure 22. Clustering result on the “rings” dataset.

4.4.3. Time Complexity of the Proposed Framework

Although using *CMNC* for the “bad data points” in our proposed framework will promote the accuracy of the clustering results, it is to some degree at the price of increasing the time complexity. Here, we will make a comparison of the execution time between the proposed framework combining different closeness measures and the traditional framework using only the Euclidean distance. The execution time of the method that directly uses *CMNC* to substitute the Euclidean distance in the traditional framework (i.e., the method without the “bad data points” identification) is also given for comparison. Meanwhile, we will also compare the execution time of the new framework under different parameter settings. All of the following tests are implemented in MATLAB R2009b, Microsoft Windows 10 operation system and based on Intel core i7 3.6-GHz CPU (quad core), 8 G DDR3 RAM.

The factors that influence the computation time of the proposed framework include the total data points number n , the clusterers number m and the number of the found “bad data points” determined by α and β . First, we will examine the change of the computation time on the synthetic dataset “double-moon” when n increases. The dataset with different numbers of data points n is shown in Figure 23.

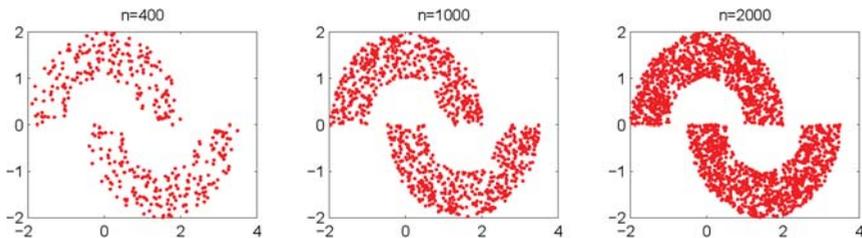


Figure 23. Synthetic dataset with different data numbers from 400–2000.

Figure 24 shows the increasing of the execution time of the three kinds of methods when the data point's number n increases. Obviously, under the shown parameter settings, the proposed framework saves much time compared to using CMNC only in the clustering method.

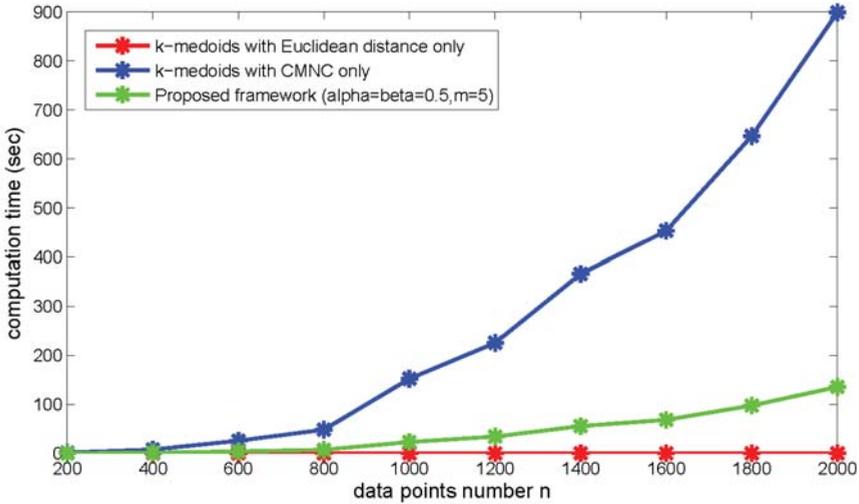


Figure 24. Increase of execution time when n increases.

Next, we will examine the change of the computation time when the clusterer's number m changes. In Figure 25, the computation time of the proposed framework using different numbers of sub-clusterers (m) is illustrated. As shown in the figure, the computation time increases along with the increasing of m . Under the shown parameter settings, if m continues to grow, the computation time of the proposed framework will exceed the time cost by the clustering method using CMNC only. In practice, m could be assigned a relatively small value in order not to produce too much computational cost.

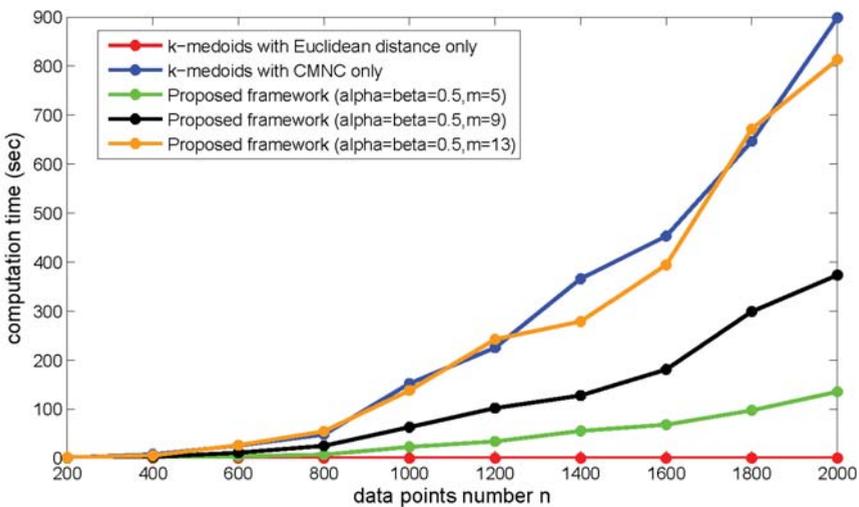


Figure 25. Increase of execution time when m increases.

The number of the found “bad data points” can also influence the computation time of the proposed framework. As shown in Figure 26, we set the data point’s number $n = 2000$ and $m = 5$, and choose different α, β values making the number of the found “bad data points” occupy from 20–50% of the total data point’s number. The figure shows that the time cost basically increases linearly with the increasing of the found “bad data points” number.

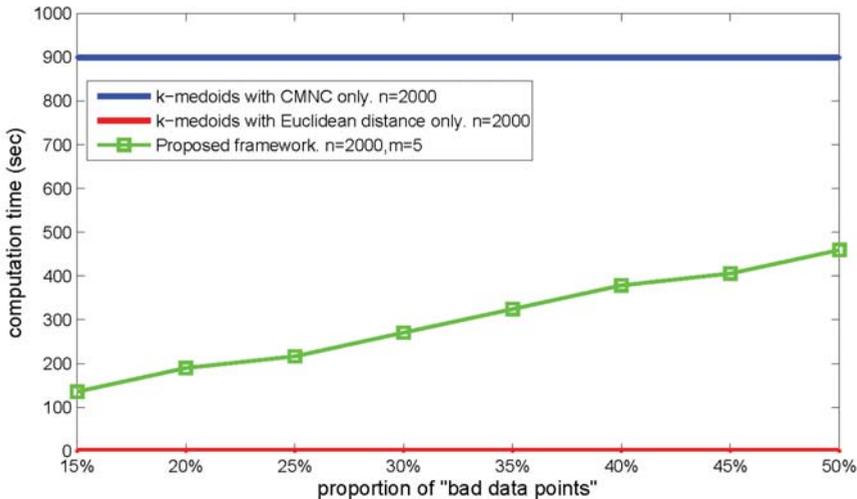


Figure 26. Increase of execution time when the number of the found “bad data points” increases.

Generally speaking, the proposed framework can generate correct clustering results on the tasks with which the traditional ones cannot deal. By using the Euclidean distance and the *CMNC* measure together, the actual usage frequency of *CMNC* can be greatly lowered. Therefore, the framework is also much more time efficient than the method using *CMNC* only.

5. Conclusions

This paper proposes a novel closeness measure between data points based on the neighborhood chain called *CMNC*. Instead of using geometric distances alone, *CMNC* measures the closeness between data points by quantifying the difficulty to establish a neighborhood chain between the two points. As shown in the experimental results, by substituting the closeness measure in traditional clustering methods, the *CMNC*-based methods can achieve much better clustering results, especially in clustering tasks with arbitrary cluster shapes and different cluster scales.

Based on *CMNC*, we also propose a multi-layer clustering ensemble framework that combines two closeness measures: the Euclidean distance and the *CMNC* metric. A “bad data points” (data points that cannot be easily grouped into the correct cluster, e.g., the data points in the overlapping regions of the two non-spherically-shaped clusters) identification algorithm is proposed to find those data points whose closeness to the clusters’ centroids need to be computed with the *CMNC* metric. By incorporating the two closeness measures, the proposed framework can counter-act the drawbacks of the traditional clustering methods using the Euclidean distance alone. Meanwhile, it is more time efficient than the clustering method using the *CMNC* metric alone.

The major focus of our future work is to further reduce the computational complexity of the proposed framework. In our work, the execution time can be reduced by limiting the number of data points that use *CMNC* metric. However, we find that in some complicated cases, in order to ensure a tolerable clustering accuracy, a large proportion of data points will be identified to be the “bad data points”. This might lead to a significant decline in the time efficiency of the

framework. To resolve the problem, more work should be done to optimize the nearest neighbors' searching process (which produces the most computational cost in CMNC) and to further refine the "bad data points" identification algorithm. One possible improvement on which we will do further research is to find some key data points that can represent part or even the whole group of the "bad data points". Applying CMNC on the representative data points might reduce the actual computation involving CMNC.

Acknowledgments: This work was supported by the National Natural Science Foundation (No. 61573275, No. 61671370), the Grant for State Key Program for Basic Research of China (973) (No. 2013CB329405), the Science and Technology Project of Shaanxi Province (No. 2013KJXX-46), the Postdoctoral Science Foundation of China (No. 2016M592790) and the Fundamental Research Funds for the Central Universities (No. xjj2014122, xjj2016066).

Author Contributions: Shaoyi Liang and Deqiang Han conceived and designed the experiments; Shaoyi Liang performed the experiments; Shaoyi Liang and Deqiang Han analyzed the data; Deqiang Han contributed analysis tools; Shaoyi Liang and Deqiang Han wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Bhatti, D.M.S.; Saeed, N.; Nam, H. Fuzzy C-Means Clustering and Energy Efficient Cluster Head Selection for Cooperative Sensor Network. *Sensors* **2016**, *16*, 1459.
- Tang, C.; Shokla, S.K.; Modhwar, G.; Wang, Q. An Effective Collaborative Mobile Weighted Clustering Schemes for Energy Balancing in Wireless Sensor Networks. *Sensors* **2016**, *16*, 261.
- Mammu, A.S.K.; Hernandez-Jayo, U.; Sainz, N.; Iglesia, I.D.L. Cross-Layer Cluster-Based Energy-Efficient Protocol for Wireless Sensor Networks. *Sensors* **2015**, *15*, 8314–8336.
- Jiang, P.; Liu, J.; Wu, F. Node Non-Uniform Deployment Based on Clustering Algorithm for Underwater Sensor Networks. *Sensors* **2015**, *15*, 29997–30010.
- Liu, G.Y.; Zhang, Y.; Wang, A. Incorporating Adaptive Local Information Into Fuzzy Clustering for Image Segmentation. *IEEE Trans. Image Proc.* **2015**, *24*, 3990–4000.
- Wang, L.F.; Pan, C.H. Robust level set image segmentation via a local correntropy-based K-means clustering. *Pattern Recognit.* **2014**, *47*, 1917–1925.
- Ji, Z.X.; Liu, J.Y.; Cao, G.; Sun, Q.S.; Chen, Q. Robust spatially constrained fuzzy c-means algorithm for brain MR image segmentation. *Pattern Recognit.* **2014**, *47*, 2454–2466.
- Yang, C.C.; Ng, T.D. Analyzing and Visualizing Web Opinion Development and Social Interactions With Density-Based Clustering. *IEEE Trans. Syst. Man Cybern. Part A* **2011**, *41*, 1144–1155.
- Huang, Q.H.; Wang, T.; Tao, D.C. Biclustering Learning of Trading Rules. *IEEE Trans. Cybern.* **2015**, *45*, 2287–2297.
- Mahmood, A.; Small, M. Subspace Based Network Community Detection Using Sparse Linear Coding. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 801–811.
- Cai, X.Y.; Li, W.J. Ranking Through Clustering: An Integrated Approach to Multi-Document Summarization. *IEEE Trans. Audio Speech Language Proc.* **2013**, *21*, 1424–1433.
- Lin, Y.S.; Jiang, J.Y.; Lee, S.J. A Similarity Measure for Text Classification and Clustering. *IEEE Trans. Knowl. and Data Eng.* **2014**, *26*, 1575–1589.
- MacQueen, J.B. Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symp. Math. Statist. Probabil.* **1967**, *1*, 281–297.
- Ester, M.; Kriegel, H.P.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, 2–4 August 1996; pp. 226–231.
- Guha, S.; Rastogi, R.; Shim, K. Cure: An Efficient Clustering Algorithm for Large Databases. *Inf. Syst.* **2001**, *26*, 35–58.
- Tzortzis, G.; Likas, A. The MinMax k-Means clustering algorithm. *Pattern Recognit.* **2014**, *47*, 2505–2516.
- Malinen, M.I.; Istodor, R.M.; Fränti, P. "K-means": Clustering by gradual data transformation. *Pattern Recognit.* **2014**, *47*, 3376–3386.
- Kumar, K.M.; Reddy, A.R.M. A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method. *Pattern Recognit.* **2016**, *58*, 39–48.

19. Morsier, F.D.; Tuia, D.; Borgeaud, M.; Gass, V.; Thiran, J.P. Cluster validity measure and merging system for hierarchical clustering considering outliers. *Pattern Recognit.* **2015**, *48*, 1478–1489.
20. Olszewski, D.; Ster, B. Asymmetric clustering using the alpha-beta divergence. *Pattern Recognit.* **2014**, *47*, 2031–2041.
21. Tabor, J.; Spurek, P. Cross-entropy clustering. *Pattern Recognit.* **2014**, *47*, 3046–3059.
22. Lu, Y.G.; Wan, Y. PHA: A fast potential-based hierarchical agglomerative clustering method. *Pattern Recognit.* **2013**, *46*, 1227–1239.
23. Chen, L.F.; Wang, S.R.; Wang, K.J.; Zhu, J.P. Soft subspace clustering of categorical data with probabilistic distance. *Pattern Recognit.* **2016**, *51*, 322–332.
24. Karypis, G.; Han, E.H.; Kumar, V. Chameleon: A Hierarchical Clustering Using Dynamic Modeling. *Computer* **1999**, *32*, 68–75.
25. Jiang, B.; Pei, J.; Tao, Y.F.; Lin, X.M. Clustering Uncertain Data Based on Probability Distribution Similarity. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 751–763.
26. Jain, A.K.; Dubes, R.C. *Algorithms for Clustering Data*; Prentice Hall: Upper Saddle River, NJ, USA, 1988.
27. Omran, M.G.; Engelbrecht, A.P.; Salman, A. An overview of clustering methods. *Intell. Data Anal.* **2007**, *11*, 583–605.
28. Lin, C.R.; Chen, M.S. Combining Partitional and Hierarchical Algorithms for Robust and Efficient Data Clustering with Cohesion Self-Merging. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 145–159.
29. Dhillon, I.S.; Maella, S.; Kumar, R. Enhanced word clustering for hierarchical text classification. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 191–200.
30. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86.
31. Dhillon, I.S.; Maella, S.; Kumar, R. A divisive information-theoretic feature clustering algorithm for text classification. *J. Mach. Learn. Res.* **2003**, *3*, 1265–1287.
32. Dhillon, I.S.; Maella, S.; Kumar, R. Information-theoretic co-clustering. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 89–98.
33. Heller, K.A.; Ghahramani, Z. Bayesian hierarchical clustering. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; Volume 22, pp. 297–304.
34. Teh, Y.W.; Li, H.D.; Roy, D. Bayesian Agglomerative Clustering with Coalescents. In Proceedings of the International Conference on Neural Information Processing Systems, Kitakyushu, Japan, 2007; pp. 1473–1480.
35. Grady, L.; Schwartz, E. Isoperimetric graph partitioning for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 496–475.
36. Pavan, M.; Pelillo, M. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 167–172.
37. Zhang, W.; Zhao, D.L.; Wang, X.G. Agglomerative clustering via maximum incremental path integral. *Pattern Recognit.* **2013**, *46*, 3056–3065.
38. Liu, Y.H.; Ma, Z.M.; Yu, F. Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy. *Knowl.-Based Syst.* **2017**, *113*, 208–220.
39. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496.
40. Sur, A.; Chowdhury, A.; Chowdhury, J.G.; Das, S. Automatic Clustering Based on Cluster Nearest Neighbor Distance (CNND) Algorithm. In Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), Bhubaneswar, India, 2013; pp. 189–196.
41. Qiu, T.; Yang, K.; Li, C.; Li, Y. A Physically Inspired Clustering Algorithm: To Evolve Like Particles. Aailbale online: <https://arxiv.org/abs/1412.5902> (accessed on 25 September 2017).
42. Qiu, T.; Li, Y. Clustering by Descending to the Nearest Neighbor in the Delaunay Graph Space. Available online: <https://arxiv.org/abs/1502.04502> (accessed on 25 September 2017).
43. Qiu, T.; Li, Y. IT-Dendrogram: A New Member of the In-Tree (IT) Clustering Family. Available online: <https://arxiv.org/abs/1507.08155> (accessed on 25 September 2017).
44. Qiu, T.; Li, Y. Clustering by Deep Nearest Neighbor Descent (D-NND): A Density-based Parameter-Insensitive Clustering Method. Available online: <https://arxiv.org/abs/1512.02097> (accessed on 25 September 2017).

45. Luo, C.; Li, Y.; Chung, S.M. Text document clustering based on neighbors. *Data Knowl. Eng.* **2009**, *68*, 1271–1288.
46. Nie, F.; Wang, X.; Huang, H. Clustering and projected clustering with adaptive neighbors. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 24–27 August 2014; pp. 977–986.
47. Liang, S.Y.; Han, D.Q.; Zhang, L.; Peng, Q.K. A Novel Clustering Oriented Closeness Measure Based on Neighborhood Chain. In Proceedings of the 2017 International Joint Conference on Neural Networks, Anchorage, AK, USA, 14–19 May 2017; pp. 997–1004.
48. Kaufmann, L.; Rousseeuw, P.J. Clustering by Means of Medoids. In *Statistical Data Analysis Based on the L1-norm & Related Methods*; North-Holland ;Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co; 1987; pp. 405–416.
49. Sneath, P.; Sokal, R. *Numerical Taxonomy*; Freeman: London, UK, 1973.
50. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering Points To Identify the Clustering Structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60.
51. Jain, A.; Law, M. Data clustering: A user’s dilemma. *Comput. Sci.* **2005**, *3776*, 1–10.
52. Zahn, C.T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.* **1971**, *100*, 68–86.
53. Chang, H.; Yeung, D.Y. Robust path-based spectral clustering. *Pattern Recognit.* **2008**, *41*, 191–203.
54. Blake, C.; Merz, C.J. UCI Repository of Machine Learning Databases. 1998. Available online: <http://www/ics.uci.edu/mllearn/MLRepository.html> (accessed on 20 May 2017).
55. Strehl, A.; Ghosh, J. Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
56. Wu, M.R.; Scholköpf, B. A Local Learning Approach for Clustering. In Proceedings of the 20th Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 2006; pp. 1529–1536.
57. Gosset, W.S. The Probable Error of a Mean. *Biometrika* **1908**, *6*, 1–25.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

A Weighted Measurement Fusion Particle Filter for Nonlinear Multisensory Systems Based on Gauss–Hermite Approximation

Yun Li ^{1,2}, Shu Li Sun ^{1,*} and Gang Hao ¹

¹ School of Electronic Engineering, Heilongjiang University, Harbin 150080, China; liyunhd@sina.com (Y.L.); haogang@hlju.edu.cn (G.H.)

² School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150001, China

* Correspondence: sunsl@hlju.edu.cn; Tel.: +86-136-7468-6865

Received: 8 August 2017; Accepted: 26 September 2017; Published: 28 September 2017

Abstract: We addressed the fusion estimation problem for nonlinear multisensory systems. Based on the Gauss–Hermite approximation and weighted least square criterion, an augmented high-dimension measurement from all sensors was compressed into a lower dimension. By combining the low-dimension measurement function with the particle filter (PF), a weighted measurement fusion PF (WMF-PF) is presented. The accuracy of WMF-PF appears good and has a lower computational cost when compared to centralized fusion PF (CF-PF). An example is given to show the effectiveness of the proposed algorithms.

Keywords: nonlinear system; weighted measurement fusion; Gauss–Hermite approximation; particle filter

1. Introduction

State estimation algorithms play an important part in automatic control, target tracking, navigation, fault diagnosis, and so on. However, it is difficult for single sensor to obtain accurate estimation and good fault tolerance, so multisensory fusion estimation technology was created to overcome this issue. There are two main kinds of fundamental estimation fusion structures: centralized fusion and distributed fusion. Centralized fusion combines measurements of all sensors into an augmented measurement, and then the data processing center outputs the fused state estimate. The advantage of centralized fusion is that there is no information loss and is ideal when all sensors are fully functional, so it is used as a comparative standard for other fusion algorithms. However, with a large number of sensors, the centralized fusion algorithm performs poorly in real time and is not reliable because of expensive computation due to augmented high-dimension measurements. In the sensor network, a large number of different types of sensors exist. If the data from the sensors are processed in a centralized way, the computational cost will be high. Especially in wireless sensor networks, large amounts of data from nodes make it difficult for decision centers to make timely decisions. In such cases, data compression must be used, and weighted measurement fusion (WMF) is one of these data compression methods [1–3].

Distributed fusion algorithms produce a fused estimate by combining local state estimates under a certain criterion in the fusion center [4–7]. Distributed fusion algorithms, such as the distributed fusion federated Kalman filter [8], the distributed weighted fusion estimation [9], and the distributed covariance intersection (CI) fusion estimation [10], are robust and flexible because of their parallel computing structures [3]. They are optimal for local application but are suboptimal for global use, as their accuracy is lower than the centralized fusion algorithm. WMF algorithms compress a high-dimension measurement to a low-dimension measurement under certain criterion [11–13], so the

computational cost can be reduced when implementing particle filter (PF) based on the compressed measurement. For linear systems, the WMF algorithms are equivalent to centralized fusion [11,13], so they are also optimal in terms of least mean squares. In this paper, the WMF method will be studied given the abovementioned advantages.

Estimation fusion has formed a complete theory for linear systems over the past years. However, most systems have nonlinear parts. For example, the measurement functions, in sensor models, of most tracking systems are established under spherical coordinates, and are strongly nonlinear when the states are estimated under Cartesian coordinates [14,15]. For nonlinear systems, fusion algorithms, which are achieved by the Taylor series, based on the Extended Kalman filter (EKF) are commonly used [16–19]. These algorithms are simple since they can be converted into linear systems. However, they result in large estimation bias and even lead to filtering divergence due to the amount of information being omitted. Many nonlinear fusion approaches have been presented, including random set, artificial neural networks, fuzzy logic, rough set, Dempster-Shafer, and other non-probabilistic approaches [20,21]. These methods fuse information and compress data. Given the amount of information lost, they are usually suboptimal.

Nonlinear filtering algorithms, based on the Bayesian estimation framework and sample approximation, have been widely studied over the past decades, including the Extended Kalman filter (EKF), Unscented Kalman Filter (UKF) [22,23], Cubature Kalman Filter (CKF), and the particle filter (PF) [24]. These algorithms are effective for nonlinear filtering problems with a single sensor [25–28]. In Hao et al. [29], we presented a weighted measurement fusion UKF (WMF-UKF) via Taylor series and UKF, which can universally handle nonlinear fusion problems. However, this algorithm needs to calculate the coefficients of a Taylor series expansion at every moment, so it is expensive and leads to slow convergence if the wrong expansion location is used.

PF can solve the estimation problem for nonlinear non-Gaussian systems and has high accuracy with an adequate number of sampling points. However, a large number of sampling points results in a huge computational burden, especially for a multisensor centralized fusion estimator. Gauss–Hermite approximation [30–33] can approximate most nonlinear functions with some sampling points and has an excellent fitting effect. We first proposed a weighted measurement fusion based on Gauss–Hermite approximation and weighted least squares. Next, a nonlinear weighted measurement fusion particle filter (WMF-PF) is presented by combining the fusion algorithm with PF for nonlinear multisensory systems. The proposed algorithm handles the nonlinear fusion problem with any noise and reduces the computational cost compared to centralized fusion PF. Moreover, it overcomes the shortcomings found in Hao et al. [29]. WMF-PF provides an effective compression method for nonlinear multisensory systems, and has potential applications in target tracking [34], communication, and massive data processing.

2. Problem Formulation

In order to facilitate the description of the algorithm, we used the scalar systems as an example. Consider the scalar nonlinear systems with L sensors:

$$x(k+1) = f(x(k), k) + w(k) \quad (1)$$

$$z^{(j)}(k) = h^{(j)}(x(k), k) + v^{(j)}(k), j = 1, 2, \dots, L \quad (2)$$

where $f(\cdot, \cdot) \in \mathfrak{R}$ is the process function, $x(k) \in \mathfrak{R}$ is the scalar state at time k , $h^{(j)}(\cdot, \cdot) \in \mathfrak{R}$ is the measurement function of the j th sensor, $z^{(j)}(k) \in \mathfrak{R}$ is the measurement of the j th sensor, $w(k) \sim p_{w_k}(\cdot)$ ($p_*(\cdot)$ is the probability density function) is the process noise, and $v^{(j)}(k) \sim p_{v_k^{(j)}}(\cdot)$ is the measurement

noise of the j th sensor. $w(k)$ and $v^{(j)}(k)$ are uncorrelated white noises with zero mean and variances, σ_w^2 and $\sigma_{v_j}^2$, i.e.,

$$\left\{ \begin{bmatrix} w(t) \\ v^{(j)}(t) \end{bmatrix} \begin{bmatrix} w^T(k) & v^{(l)T}(k) \end{bmatrix} \right\} = \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_{v_j}^2 \delta_{jl} \end{bmatrix} \delta_{tk} \quad (3)$$

where E denotes the mathematical expectation, the superscript T denotes the transpose, and δ_{tk} and δ_{jl} are the Kronecker delta functions, i.e., $\delta_{tt} = 1$ and $\delta_{tk} = 0 (t \neq k)$.

Assumption 1: The $f(\cdot, \cdot)$, $h^{(j)}(\cdot, \cdot)$, $p_{w_k}(\cdot)$, and $p_{v_k^{(j)}}(\cdot)$ are known.

Assumption 2: The state $x(k)$ is bounded.

For the systems in Equations (1) and (2), the augmented measurement equation of the centralized fusion system (CFS) is given a

$$z^{(0)}(k) = \mathbf{h}^{(0)}(x(k), k) + \mathbf{v}^{(0)}(k) \quad (4)$$

where

$$z^{(0)}(k) = [z^{(1)}(k), z^{(2)}(k), \dots, z^{(L)}(k)]^T \quad (5)$$

$$\mathbf{h}^{(0)}(x(k), k) = [h^{(1)}(x(k), k), h^{(2)}(x(k), k), \dots, h^{(L)}(x(k), k)]^T \quad (6)$$

$$\mathbf{v}^{(0)}(k) = [v^{(1)}(k), v^{(2)}(k), \dots, v^{(L)}(k)]^T \quad (7)$$

and the covariance matrix of $\mathbf{v}^{(0)}(k)$ is given as

$$\mathbf{R}^{(0)} = \text{diag}(\sigma_{v1}^2, \sigma_{v2}^2, \dots, \sigma_{vL}^2) \quad (8)$$

where $\text{diag}(\cdot)$ denotes a diagonal matrix.

For the systems in Equations (1) and (4), we obtained the centralized fusion PF combined with the particle filter. However, Equation (4), with a high dimension, will result in high computational costs, particularly in massive sensor networks. Therefore, it is important to find the equivalent or approximate fusion methods to reduce the computational cost.

Lemma 1. For the systems in Equations (1) and (2), if there are linear relationships between measurement functions $h^{(j)}(x(k), k)$, $j = 1, 2, \dots, L$, that is to say, there is a intermediary function $\mathbf{h}(x(k), k) \in \mathbb{R}^{p \times 1}$ that satisfies $h^{(j)}(x(k), k) = \mathbf{H}^{(j)}(k)\mathbf{h}(x(k), k)$ with matrix $\mathbf{H}^{(j)}(k) \in \mathbb{R}^{1 \times p}$, the compressed measurement function of the weighted measurement fusion system (WMFS) is given as [29]

$$z^{(1)}(k) = \mathbf{H}^{(1)}(k)\mathbf{h}(x(k), k) + \mathbf{v}^{(1)}(k) \quad (9)$$

$$z^{(1)}(k) = [\mathbf{M}^T(k)\mathbf{R}^{(0)-1}\mathbf{M}(k)]^{-1}\mathbf{M}^T(k)(\mathbf{R}^{(0)})^{-1}z^{(0)}(k) \quad (10)$$

$$\mathbf{v}^{(1)}(k) = [\mathbf{M}^T(k)\mathbf{R}^{(0)-1}\mathbf{M}(k)]^{-1}\mathbf{M}^T(k)(\mathbf{R}^{(0)})^{-1}\mathbf{v}^{(0)}(k) \quad (11)$$

$$\mathbf{R}^{(0)-1} = (\mathbf{R}^{(0)})^{-1} \quad (12)$$

The covariance matrix of $\mathbf{v}^{(1)}(k)$ is computed by

$$\mathbf{R}^{(1)}(k) = [\mathbf{M}^T(k)\mathbf{R}^{(0)-1}\mathbf{M}(k)]^{-1} \quad (13)$$

where $\mathbf{M}(k)$, with full-column rank, and $\mathbf{H}^{(1)}(k)$, with full-row rank, are the full rank decomposition matrices of matrix $\mathbf{H}^{(0)}(k) = [\mathbf{H}^{(1)T}(k), \dots, \mathbf{H}^{(L)T}(k)]^T$, that is,

$$\mathbf{H}^{(0)}(k) = \mathbf{M}(k)\mathbf{H}^{(1)}(k) \quad (14)$$

which can be computed by Hermite canonical form [29].

We assumed that the statistics of $w(k)$ and $v^{(j)}(k), j = 1, 2, \dots, L$ are known. In fact, we obtained the noise statistics through identification [35–37]. For time-invariant systems, we identified them offline and obtained the optimal weighted measurement fusion algorithms. For time-varying systems, we identified them online and obtained the asymptotic optimal adaptive weighted measurement fusion algorithm.

In order to determine the weighted measurement fusion based on the probabilistic method, we needed to know two things: the relationships among the noise statistics of all the sensors and the relationships among the measurement functions of all sensors. For systems with additive noises, if we knew the relationships between the measurement functions, the relationships among the measurement noise statistics could be determined. If a linear relationship exists among the measurement functions, whether the measurement functions themselves are linear or not, we could use the least square method to find the optimal compressed measurement. If a nonlinear relationship exists among the measurement functions, we could use intermediary functions to achieve the optimal compressed measurement. Unfortunately, finding the intermediary function is difficult because of the complexity and diversity of nonlinear functions.

For example, there are four sensors and their measurement functions are as follows:

$$\begin{aligned} z^{(1)}(k) &= 1 + x(k) + x^2(k) + v^{(1)}(k), z^{(2)}(k) = 2 + x(k) + 2x^2(k) + x^3(k) + v^{(2)}(k), \\ z^{(3)}(k) &= 3 + 2x(k) + 3x^2(k) + x^3(k) + v^{(3)}(k), z^{(4)}(k) = 4 + x(k) + 4x^2(k) + 3x^3(k) + v^{(4)}(k) \end{aligned} \tag{15}$$

where the variances of $v^{(j)}(k), j = 1 \dots 4$ are $R^{(1)} = 1, R^{(2)} = 2, R^{(3)} = 3,$ and $R^{(4)} = 4.$

Set $h(x(k), k) = [1 \ x(k) \ x^2(k) \ x^3(k)]^T,$ then Equation (15) can be written as

$$\begin{aligned} z^{(1)}(k) &= [1 \ 1 \ 1 \ 0]h(x(k), k) + v^{(1)}(k), z^{(2)}(k) = [2 \ 1 \ 2 \ 1]h(x(k), k) + v^{(2)}(k), \\ z^{(3)}(k) &= [3 \ 2 \ 3 \ 1]h(x(k), k) + v^{(3)}(k), z^{(4)}(k) = [4 \ 1 \ 4 \ 3]h(x(k), k) + v^{(4)}(k) \end{aligned} \tag{16}$$

Let

$$H^{(0)} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 2 & 1 \\ 3 & 2 & 3 & 1 \\ 4 & 1 & 4 & 3 \end{bmatrix} = MH^{(1)} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 2 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \tag{17}$$

where M is full-column rank, and $H^{(1)}$ is full-row rank. According to the weighted least squares criterion, the fused measurement function is

$$z^{(1)}(k) = H^{(1)}(k)h(x(k), k) + v^{(1)}(k) \tag{18}$$

where $z^{(1)}(k) = [M^T(k)R^{(0)}]^{-1}M(k)]^{-1}M^T(k)R^{(0)}]^{-1}z^{(0)}(k) = \begin{bmatrix} -0.3286 & 0.1000 & -0.0429 & 0.3143 \\ 0.8571 & 0 & 0.2857 & -0.4286 \end{bmatrix} z^{(0)}(k),$

and the covariance of $v^{(1)}(k)$ is $R^{(1)}(k) = [M^T(k)R^{(0)}]^{-1}M(k)]^{-1} = \begin{bmatrix} 0.5286 & -0.8571 \\ -0.8571 & 1.7143 \end{bmatrix}.$

The measurement function of WMF is two-dimensional, and that of the centralized fusion system is four-dimensional, so the WMF can compress measurement dimensions effectively and reduce the computational cost.

In the next section, we provide an approximation method to solve this problem.

3. Gauss–Hermite Approximation

Lemma 2. Let $y(x)$ be a determined function [30]. Assuming that there is an ensemble of S points $\{x_i, i = 1, \dots, S\}$ distributed uniformly in the interval $[a, b],$ there exists a point y_i such that $y_i = y(x_i)$ for each point $x_i.$ Then, the approximation function $\bar{y}(x)$ of $y(x),$ by Gauss–Hermite folding reads

$$\bar{y}(x) = \frac{1}{\gamma\sqrt{\pi}} \sum_{i=1}^S y_i \Delta x_i \exp \left\{ - \left(\frac{x - x_i}{\gamma} \right)^2 \right\} f_n \left(\frac{x - x_i}{\gamma} \right) \tag{19}$$

where γ is a coefficient, $\Delta x_i = \frac{1}{2}(x_{i+1} - x_{i-1})$, and $f_n(u)$ is the correction polynomials which can be decomposed into a series of Hermite polynomials:

$$f_n(u) = \sum_{i=1}^n C_i H_i(u) \tag{20}$$

$$C_i = \frac{1}{2^i i!} H_i(0) \tag{21}$$

where $H_i(u) = (-1)^i e^{u^2} (e^{-u^2})^{(i)}$ represents Hermite polynomials [31], and $H_i(0)$ is defined as

$$H_i(0) = \begin{cases} 1 & i = 0 \\ 2^n (-1)^n (2n - 1)!! & i = 2n \\ 0 & i = 2n + 1 \end{cases} \tag{22}$$

From Equations (21) and (22), we obtain

$$C_i = \begin{cases} 1 & i = 0 \\ (-1)^n \frac{(2n-1)!!}{2^n (2n)!} & i = 2n \\ 0 & i = 2n + 1 \end{cases} \tag{23}$$

See the detailed proof in the literature [30–33].

Remark 1. Although γ is an arbitrary coefficient related to Δx_i , it was found that $\bar{y}(x)$ approximates better when $\gamma = 1$ and $n = 2$ or 4 , for most elementary functions [30]. Thus, we used $\gamma = 1$ and $n = 2$ in this paper, which yielded

$$\bar{y}(x) = \frac{1}{\sqrt{\pi}} \sum_{i=1}^S y_i \Delta x_i e^{-(x-x_i)^2} (1.5 - (x - x_i)^2) \tag{24}$$

4. Universal WMF Based on Gauss–Hermite Approximation

If the function $\sum_{i=1}^S e^{-(x-x_i)^2} (1.5 - (x - x_i)^2)$ in Equation (24) is seen as the intermediary function $h(x(k), k)$, and $\frac{1}{\sqrt{\pi}} \sum_{i=1}^S y_i \Delta x_i$ is seen as $H^{(j)}(k)$ in Lemma 2, then we can obtain the linear relationships among the measurement functions $h^{(j)}(x(k), k)$ and obtain the following theorem using Lemma 2.

Based on Lemmas 1 and 2, we can obtain the following Theorem 1.

Theorem 1. For the systems in Equations (1) and (2), and from Equation (24), the approximate measurement equation of the weighted measurement fusion system is

$$\bar{z}^{(1)}(k) = \bar{H}^{(1)} \begin{pmatrix} e^{-(x-x_1)^2} (1.5 - (x - x_1)^2) \\ e^{-(x-x_2)^2} (1.5 - (x - x_2)^2) \\ \vdots \\ e^{-(x-x_S)^2} (1.5 - (x - x_S)^2) \end{pmatrix} + \bar{v}^{(1)}(k) \tag{25}$$

We denoted

$$\bar{\mathbf{H}}^{(0)} = \frac{1}{\sqrt{\pi}} \begin{pmatrix} y_1^{(1)} \Delta x_1^{(1)} & y_2^{(1)} \Delta x_2^{(1)} & \cdots & y_S^{(1)} \Delta x_S^{(1)} \\ y_1^{(2)} \Delta x_1^{(2)} & y_2^{(2)} \Delta x_2^{(2)} & \cdots & y_S^{(2)} \Delta x_S^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ y_1^{(L)} \Delta x_1^{(L)} & y_2^{(L)} \Delta x_2^{(L)} & \cdots & y_S^{(L)} \Delta x_S^{(L)} \end{pmatrix} \quad (26)$$

$$\bar{\mathbf{z}}^{(1)}(k) = [\bar{\mathbf{M}}^T \mathbf{R}^{(0)-1} \bar{\mathbf{M}}]^{-1} \bar{\mathbf{M}}^T \mathbf{R}^{(0)-1} \mathbf{z}^{(0)}(k) \quad (27)$$

$$\bar{\mathbf{v}}^{(1)}(k) = [\bar{\mathbf{M}}^T \mathbf{R}^{(0)-1} \bar{\mathbf{M}}]^{-1} \bar{\mathbf{M}}^T \mathbf{R}^{(0)-1} \mathbf{v}^{(0)}(k) \quad (28)$$

The covariance matrix of $\mathbf{v}^{(1)}(k)$ is given by

$$\bar{\mathbf{R}}^{(1)} = [\bar{\mathbf{M}}^T \mathbf{R}^{(0)-1} \bar{\mathbf{M}}]^{-1} \quad (29)$$

where $y_k^{(m)}(k = 1, \dots, S)$ is the k th sample point of the m th sensor, S is the sample point number, $\bar{\mathbf{M}}$ and $\bar{\mathbf{H}}^{(1)}$ are the full rank decomposition matrices of $\bar{\mathbf{H}}^{(0)}$, $\bar{\mathbf{M}} \in \mathbf{R}^{L \times r}$ is full-column rank, $\bar{\mathbf{H}}^{(1)} \in \mathbf{R}^{r \times S}$ is full-row rank, $r = \text{rank}(\bar{\mathbf{H}}^{(0)})$, and $r \leq \min\{L, S\}$.

An approximate intermediary function,

$$\mathbf{h}(x(k), k) = \begin{pmatrix} e^{-(x-x_1)^2} (1.5 - (x-x_1)^2) \\ e^{-(x-x_2)^2} (1.5 - (x-x_2)^2) \\ \vdots \\ e^{-(x-x_S)^2} (1.5 - (x-x_S)^2) \end{pmatrix} \quad (30)$$

has been constructed using the Gauss–Hermite approximation method in Lemma 2. It establishes some linear relationships between local measurement functions. Therefore, it solves the restriction that the measurement functions must be linear relationships in Lemma 1.

From the above theorem, we can see that, if there is a nonlinear system with L sensors, the centralized fusion system in Equations (1) and (4) has a measurement function with L dimension and the WMF system in Equations (1) and (25) has a measurement function with r dimension. Because $r = \text{rank}(\bar{\mathbf{H}}^{(0)})$ and $r \leq \min\{L, S\}$, the computational cost of the WMF filter is less than that of the centralized fusion filter. After the sample point number S is determined, the compression efficiency of the algorithm is significantly improved if there are a large number of sensors ($L \gg S$ and S is a constant).

5. WMF-PF Based on Gauss–Hermite Approximation

In this section, we propose a WMF-PF algorithm based on the WMF system in Equations (1) and (25) with a low-dimension measurement.

5.1. WMF-PF Algorithm

The computational procedure of WMF-PF is given as follows:

1. Initialization: $\hat{\mathbf{x}}^{(1)(i)}(0|0) \sim p_{x_0}(x_0)$, $i = 1, \dots, N_s$;
2. State prediction particles:

$$\hat{\mathbf{x}}^{(1)(i)}(k|k-1) = f(\hat{\mathbf{x}}^{(1)(i)}(k-1|k-1), k-1) + \zeta^{(1)(i)}(k-1) \quad (31)$$

where $\zeta^{(1)(i)}(k-1)$ is random number with the same distribution of the process noise $w(k)$;

3. Measurement prediction particles:

$$\hat{z}^{(1)(i)}(k|k-1) = \bar{H}^{(1)} h(\hat{x}^{(1)(i)}(k|k-1), k) \tag{32}$$

4. The importance weight:

$$\omega_k^{(1)(i)} = \frac{1}{N_s} p(\hat{z}^{(1)(i)}(k|k-1) | \hat{x}^{(1)(i)}(k|k-1)) \tag{33}$$

that is,

$$\omega_k^{(1)(i)} = \frac{1}{N_s} p_{v_k^{(1)}}(\bar{z}^{(1)}(k) - \hat{z}^{(1)(i)}(k|k-1)) \tag{34}$$

where $\bar{z}^{(1)}(k)$ is computed by Equation (27), and $\bar{\omega}_k^{(1)(i)}$ is given by

$$\bar{\omega}_k^{(1)(i)} = \frac{\omega_k^{(1)(i)}}{\sum_{i=1}^N \omega_k^{(1)(i)}} \tag{35}$$

5. Filtering:

$$\hat{x}^{(1)}(k|k) = \sum_{i=1}^{N_s} \bar{\omega}_k^{(1)(i)} \hat{x}^{(1)(i)}(k|k-1) \tag{36}$$

and the filtering variance matrix is computed by

$$P^{(1)}(k|k) \approx \sum_{i=1}^{N_s} \bar{\omega}_k^{(1)(i)} (\hat{x}^{(1)(i)}(k|k-1) - \hat{x}^{(1)(i)}(k|k))^2 \tag{37}$$

6. Resampling: In this paper, systematic sampling is used as the resampling method, i.e.,

$$u_i = \frac{(i-1) + r}{N}, r \sim U[0, 1], i = 1, \dots, N_s \tag{38}$$

If $\sum_{j=1}^{m-1} \bar{\omega}_k^{(1)(j)} < u_i \leq \sum_{j=1}^m \bar{\omega}_k^{(1)(j)}$, we directly copied m particles as the resampling particles $\hat{x}^{(1)(i)}(k|k)$.

We then returned to Step 2 and reiterated.

The flow chart of the WMF-PF algorithm is shown in Figure 1.

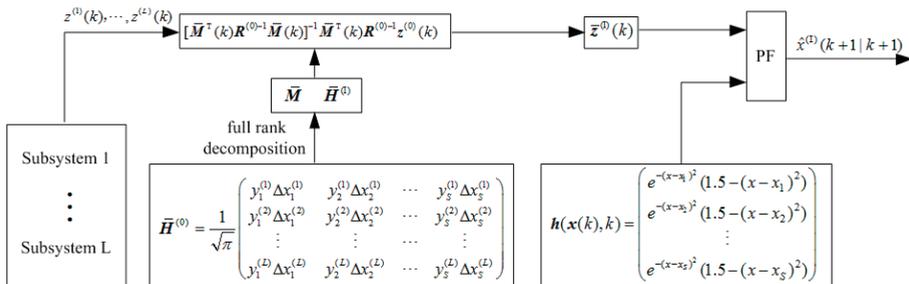


Figure 1. Flow chart of the weighted measurement fusion particle filter (WMF-PF) algorithm.

5.2. Time Complexity Analysis

From Equations (31)–(38), we found that the time complexity is determined by Equation (32). From Equation (32), the time complexity of centralized fusion PF (CF-PF) is $O(L^2)$ and that of WMF-PF is $O(r^2)$. Because $r \leq \min\{L, S\}$, the time complexity of WMF-PF is less than that of CMF-PF. Particularly, when there are a large number of sensors ($L \gg S$), the computational cost can be substantially reduced.

6. Simulation Examples

6.1. Model Description

Let us consider a classical 10-sensor nonlinear system [38]:

$$x(k) = \frac{x(k-1)}{4} + x(k-1)/(1+x(k-1)^2) + 2\cos(0.5(k-1)) + w(k) \tag{39}$$

$$z^{(j)}(k) = h^{(j)}(x(k), k) + v^{(j)}(k), j = 1, \dots, 10 \tag{40}$$

Taking observability into account, the sensors selected were single-valued functions within the range of $x(k)$:

$$\begin{aligned} h^{(1)}(x(k), k) &= 0.8x(k) + v^{(1)}(k) \\ h^{(2)}(x(k), k) &= 1.2x(k) + v^{(2)}(k) \\ h^{(3)}(x(k), k) &= \exp(x(k)/3) + v^{(3)}(k) \\ h^{(4)}(x(k), k) &= 1.2 \exp(x(k)/3) + v^{(4)}(k) \\ h^{(5)}(x(k), k) &= 0.05x(k)^3 + v^{(5)}(k) \\ h^{(6)}(x(k), k) &= 0.06x(k)^3 + v^{(6)}(k) \\ h^{(7)}(x(k), k) &= 5 \sin(0.1\pi x(k)) + v^{(7)}(k) \\ h^{(8)}(x(k), k) &= 6 \sin(0.1\pi x(k)) + v^{(8)}(k) \\ h^{(9)}(x(k), k) &= 5 \arctan(0.1\pi x(k)) + v^{(9)}(k) \\ h^{(10)}(x(k), k) &= 6 \arctan(0.1\pi x(k)) + v^{(10)}(k) \end{aligned} \tag{41}$$

The $w(k) \sim U(0, 1)$ is uniformly distributed, and $v^{(j)}(k), j = 1, \dots, 10$ are uncorrelated Gaussian noises with variance $\sigma_{v_j}^2 = (0.5 + 0.01j)^2$. The initial state is $x(0) = 0$.

6.2. Gauss–Hermite Approximation

Because the state $x(k)$ ranges from -3 to 4 , we chose $S = 10$ sampling points ($x_i = -4, -3, \dots, 5$) to approximate, and the corresponding coefficients were $\gamma = 1, \Delta x_i = 1$, and $n = 2$. Their mean square errors (MSEs) are shown in Table 1, and the approximation curves are shown in Figure 2.

Table 1. Mean square errors.

Sensor Functions	$h^{(1)}(\cdot)$	$h^{(2)}(\cdot)$	$h^{(3)}(\cdot)$	$h^{(4)}(\cdot)$	$h^{(5)}(\cdot)$	$h^{(6)}(\cdot)$	$h^{(7)}(\cdot)$	$h^{(8)}(\cdot)$	$h^{(9)}(\cdot)$	$h^{(10)}(\cdot)$
MSEs using Gauss–Hermite	0.0032	0.0017	0.0010	0.0014	0.0029	0.0042	0.0009	0.0013	0.0010	0.0015
MSEs using McLaughlin series	0	0	0.0258	0.0371	0.9621	1.3854	0.2286	0.3292	0.3963	0.5707

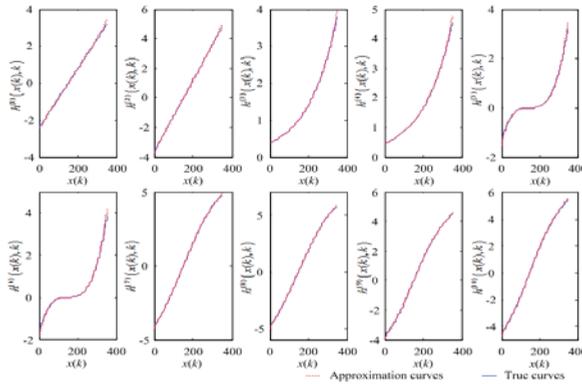


Figure 2. Approximation curves of nonlinear functions.

To compare with other approximate methods, we introduced the approximation method using a McLaughlin series. We used a third-order McLaughlin series to approximate the nonlinear functions $h^{(j)}(x(k), k)$. The MSEs of the approximation algorithm, using a McLaughlin series, are also shown in Table 1. From Table 1, we can see that the MSEs of the approximation algorithm using a McLaughlin series are larger than that when using Gauss–Hermite approximation.

6.3. Estimation Using WMF-PF

From the above experiments, we see that the approximation effect is good when using the above coefficients. Next, we established the measurement equation and fusion matrix of WMF. As the intermediary function $h(x(k), k)$ is $(e^{-(x-x_1)^2}(1.5 - (x - x_1)^2), \dots, e^{-(x-x_{10})^2}(1.5 - (x - x_{10})^2))^T$, from Equations (24) and (25), the coefficient matrices $\bar{H}^{(0)}$, and its full rank decomposition matrices \bar{M} and $\bar{H}^{(1)}$, can be computed as follows:

$$\begin{aligned}
 \bar{H}^{(0)} &= \begin{bmatrix} -1.8054 & -1.3541 & -0.9027 & -0.4514 & 0 & 0.4514 & 0.9027 & 1.3541 & 1.8054 & 2.2568 \\ -2.7081 & -2.0311 & -1.3541 & -0.6770 & 0 & 0.6770 & 1.3541 & 2.0311 & 2.7081 & 3.3851 \\ 0.1487 & 0.2076 & 0.2897 & 0.4043 & 0.5642 & 0.7874 & 1.0989 & 1.5336 & 2.1403 & 2.9871 \\ 0.1785 & 0.2491 & 0.3476 & 0.4851 & 0.6770 & 0.9449 & 1.3187 & 1.8404 & 2.5684 & 3.5845 \\ -1.8054 & -0.7617 & -0.2257 & -0.0282 & 0 & 0.0282 & 0.2257 & 0.7617 & 1.8054 & 3.5262 \\ -2.1665 & -0.9140 & -0.2708 & -0.0339 & 0 & 0.0339 & 0.2708 & 0.9140 & 2.1665 & 4.2314 \\ -2.6829 & -2.2822 & -1.6581 & -0.8717 & 0 & 0.8717 & 1.6581 & 2.2822 & 2.6829 & 2.8209 \\ -3.2195 & -2.7386 & -1.9897 & -1.0461 & 0 & 1.0461 & 1.9897 & 2.7386 & 3.2195 & 3.3851 \\ -2.5350 & -2.1321 & -1.5825 & -0.8587 & 0 & 0.8587 & 1.5825 & 2.1321 & 2.5350 & 2.8319 \\ -3.0420 & -2.5585 & -1.8990 & -1.0304 & 0 & 1.0304 & 1.8990 & 2.5585 & 3.0420 & 3.3983 \end{bmatrix} \\
 \bar{M} &= \begin{bmatrix} -1.8054 & -1.3541 & -0.9027 & -0.4514 & 0 \\ -2.7081 & -2.0311 & -1.3541 & -0.6770 & 0 \\ 0.1487 & 0.2076 & 0.2897 & 0.4043 & 0.5642 \\ 0.1785 & 0.2491 & 0.3476 & 0.4851 & 0.6770 \\ -1.8054 & -0.7617 & -0.2257 & -0.0282 & 0 \\ -2.1665 & -0.9140 & -0.2708 & -0.0339 & 0 \\ -2.6829 & -2.2822 & -1.6581 & -0.8717 & 0 \\ -3.2195 & -2.7386 & -1.9897 & -1.0461 & 0 \\ -2.5350 & -2.1321 & -1.5825 & -0.8587 & 0 \\ -3.0420 & -2.5585 & -1.8990 & -1.0304 & 0 \end{bmatrix} \\
 \bar{H}^{(1)} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -6.0696 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 15.5971 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & -21.6922 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 15.8716 \\ 0 & 0 & 0 & 0 & 1 & 2.1121 & 2.4612 & 3.0862 & 4.0573 & 0.9212 \end{pmatrix}.
 \end{aligned} \tag{42}$$

Using the WMF-PF algorithm, we obtained the state estimate for the system in Equations (39) and (40). The curves of the true values and their estimates, using WMF-PF based on Gauss–Hermite approximation, are shown in Figure 3. The tracking performance is optimal.

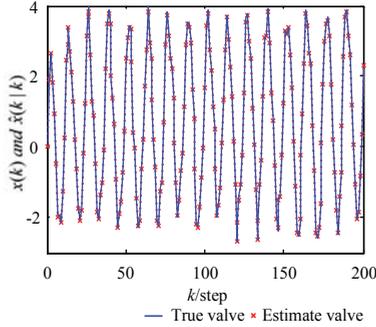


Figure 3. Curves of the true values and estimates using the WMF-PF algorithm based on Gauss–Hermite approximation.

6.4. Analysis

In order to compare our findings with other fusion algorithms, we introduced a kind of distributed fusion method, called a fast covariance intersection (CI) fusion algorithm [39]. Its calculation process is

$$\hat{x}_{k|k}^{(CI)} = \sum_{i=1}^L \omega_i P_{CI} P_i^{-1} \hat{x}_{k|k}^{(j)} \tag{43}$$

where $\hat{x}_{k|k}^{(CI)}$ is the fusion estimate by CI, $\hat{x}_{k|k}^{(j)}$, $i = 1, \dots, L$ are the estimates of subsystems, P_i , $i = 1, \dots, L$ are the filter error variance matrices of subsystems, and

$$P_{CI}^{-1} = \sum_{i=1}^L \omega_i P_i^{-1} \tag{44}$$

$$\omega_j = \frac{\text{trace}(P_j^{-1})}{\sum_{i=1}^L \text{trace}(P_i^{-1})} \tag{45}$$

Using the CI fusion algorithm and the PF algorithm, we obtained the covariance intersection PF (CI-PF). This fusion algorithm is simple, robust, and flexible. However, its accuracy is lower than WMF. We introduced an evaluation indicator called the accumulated mean square error (AMSE) [29]:

$$\text{AMSE}(k) = \sum_{t=0}^k \frac{1}{N} \sum_{j=1}^N [x(t) - \hat{x}_j(t|t)]^2 \tag{46}$$

where $\hat{x}_j(t|t)$ is the j th-time Monte Carlo experiment at time t . The AMSE curves of local PFs (LF1–LF7), the WMF-PF, and the CI-PF are shown in Figure 4 with 20 Monte Carlo experiments. From Figure 4, the WMF-PF based on Gauss–Hermite approximation is shown to have better accuracy than local PF and CI-PF.

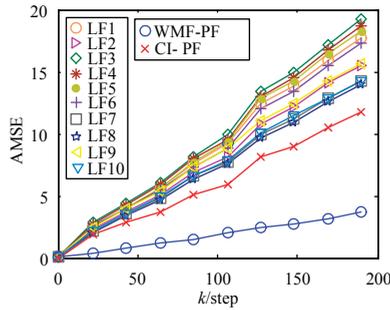


Figure 4. Accumulated mean square error (AMSE) curves of local particle filters (PFs), WMF-PF, and covariance intersection PF (CI-PF).

In addition, we simulated the AMSE curves of WMF-PF under $S = 8, x_i = -3.6, -2.4, \dots, 4.8, \Delta x_i = 1.2, n = 2,$ and $\gamma = 1$ and WMF-PF under $S = 10, \Delta x_i = 1, n = 2,$ and $\gamma = 0.9$. To distinguish them with the above WMF-PF, we named them WMF-PF-1 and WMF-PF-2. The AMSE curves of WMF-PF, WMF-PF-1, and WMF-PF-2 are shown in Figure 5. We can see that, as the interval Δx_i becomes larger, the computational cost is reduced, but the estimation accuracy declines, and a reasonable parameter γ can improve the estimation accuracy. The AMSE curves of WMF-PF and CF-PF are shown in Figure 6. We can see that the accuracy of WMF-PF approximates that of CF-PF.

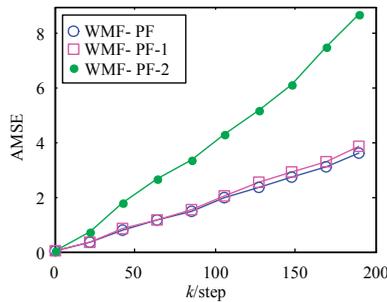


Figure 5. AMSE curves of WMF-PF, WMF-PF-1, and WMF-PF-2.

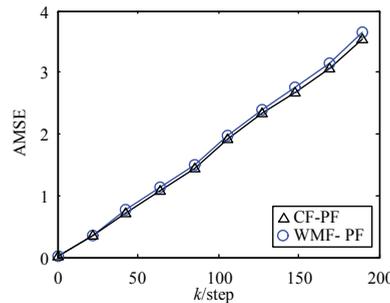


Figure 6. AMSE curves of WMF-PF and CF-PF.

Next, we compare the computational cost of WMP-PF and CF-PF. Because $\mathbf{H}^{(1)}$ is a 5×10 matrix, the time complexity of WMP-PF is $O(5^2)$. However, $\mathbf{H}^{(0)}$ is a 10×10 matrix and its time complexity is $O(10^2)$. Therefore, the computational cost of WMF-PF is obviously lower than that of CF-PF.

In summary, the proposed WMP-PF, based on Gauss–Hermite approximation, is more accurate compared with local PFs, and has a lower computation cost when compared with CF-PF.

7. Conclusions

A general WMF algorithm is presented here for nonlinear multisensory systems based on the Gauss–Hermite approximation of nonlinear functions and the weighted least square method. An augmented high-dimension measurement from all sensors is compressed to a low-dimension one. Combined with the particle filter, the nonlinear weighted measurement fusion PF (WMF-PF) is presented. It can handle the nonlinear fusion estimation problem, and has reasonable accuracy and a reduced computational cost compared to centralized fusion PF. The proposed WMF algorithm can be used in systems of large-scale sensors, which can significantly reduce the computational cost and improve the real-time feature.

Acknowledgments: This work was supported by the Natural Science Foundation of China (No. 61573132 and No. 61503127), by the Natural Science Foundation of Heilongjiang province (No. F2015014), by Chang Jiang Scholar Candidates Program for Provincial Universities in Heilongjiang (No. 2013CJHB005), by the Science and Technology Innovative Research Team in Higher Educational Institutions of Heilongjiang Province (No. 2012TD007), and by the Electronic Engineering Province Key Laboratory.

Author Contributions: Shuli Sun proposed the algorithm. Yun Li derived the algorithm and wrote the paper. Gang Hao performed the simulation work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gispan, L.; Leshem, A.; Be’ery, Y. Decentralized estimation of regression coefficients in sensor networks. *Digit. Signal Proc.* **2017**, *68*, 16–23. [CrossRef]
2. Song, E.B.; Zhu, Y.M.; Zhou, J. Sensors’ optimal dimensionality compression matrix in estimation fusion. *Automatica* **2005**, *41*, 2131–2139. [CrossRef]
3. Sun, S.; Lin, H.; Ma, J.; Li, X. Multi-sensor distributed fusion estimation with applications in networked systems: A review paper. *Inf. Fusion* **2017**, *38*, 122–134. [CrossRef]
4. Sun, S.L.; Deng, Z.L. Multi-sensor optimal information fusion Kalman filter. *Automatica* **2004**, *40*, 1017–1023. [CrossRef]
5. Ma, J.; Sun, S.L. Information fusion estimators for systems with multiple sensors of different packet dropout rates. *Inf. Fusion* **2011**, *12*, 213–222. [CrossRef]
6. Tian, T.; Sun, S.L.; Li, N. Multi-sensor information fusion estimators for stochastic uncertain systems with correlated noises. *Inf. Fusion* **2016**, *27*, 126–137. [CrossRef]
7. Ge, Q.B.; Shao, T.; Duan, Z.S.; Wen, C.L. Performance Analysis of the Kalman Filter with Mismatched Measurement Noise Covariance. *IEEE Trans. Autom. Control* **2016**, *61*, 4014–4019. [CrossRef]
8. Carlson, N. Federated square root filter for decentralized parallel processes. *IEEE Trans. Aerosp. Electron. Syst.* **1990**, *26*, 517–525. [CrossRef]
9. Kim, K. Development of track to track fusion algorithm. In Proceedings of the American Control Conference, Baltimore, MD, USA, 29 June–1 July 1994.
10. Julier, S.; Uhlmann, J. A non-divergent estimation algorithm in the presence of unknown correlation. In Proceedings of the American Control Conference, Albuquerque, NM, USA, 6 June 1997; pp. 2369–2373.
11. Qiang, G.; Harris, C.J. Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion. *IEEE Trans. Aerosp. Electron. Syst.* **2001**, *37*, 273–280. [CrossRef]
12. Oussalah, M.; Messaoudi, Z.; Ouldali, A. Track-to-Track Measurement Fusion Architectures and Correlation Analysis. *J. Univers. Comput. Sci.* **2010**, *16*, 37–61.
13. Ran, C.J.; Deng, Z.L. Self-tuning weighted measurement fusion Kalman filtering algorithm. *Comput. Stat. Data Anal.* **2012**, *56*, 2112–2128. [CrossRef]

14. Li, X.R.; Jilkov, V.P. A Survey of Maneuvering Target Tracking: Approximation Techniques for Nonlinear Filtering. *Proc. SPIE* **2004**, *5428*, 537–550. [CrossRef]
15. Bar-Shalom, Y.; Li, X.R. *Multitarget-Multisensor Tracking: Principles and Techniques*; YBS Publishing: Storrs, CT, USA, 1995.
16. Assa, A.; Janabi-Sharifi, F. A Kalman Filter-Based Framework for Enhanced Sensor Fusion. *IEEE Sens. J.* **2015**, *15*, 3281–3292. [CrossRef]
17. Straka, O.; Duník, J.; Šimandl, M. Truncation nonlinear filters for state estimation with nonlinear inequality constraints. *Automatica* **2012**, *48*, 273–286. [CrossRef]
18. Hlinka, O.; Slučiak, O.; Hlawatsch, F.; Djuric, P.M.; Rupp, M. Likelihood consensus and its application to distributed particle filtering. *IEEE Trans. Signal Proc.* **2012**, *60*, 4334–4349. [CrossRef]
19. Ge, Q.B.; Xu, D.X.; Wen, C.L. Cubature information filters with correlated noises and their applications in decentralized fusion. *Signal Proc.* **2014**, *94*, 434–444. [CrossRef]
20. Jia, B.; Xin, M.; Cheng, Y. High-degree cubature Kalman filter. *Automatica* **2013**, *49*, 510–518. [CrossRef]
21. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *14*, 28–44. [CrossRef]
22. Julier, S.J.; Uhlmann, J.K. A new extension of the Kalman filter to nonlinear systems. In Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL, USA, 28 July 1997.
23. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [CrossRef]
24. Arulampalam, S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Proc.* **2002**, *50*, 174–188. [CrossRef]
25. Chatzi, E.N.; Smyth, A.W. The unscented Kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing. *Struct. Control Health Monit.* **2009**, *16*, 99–123. [CrossRef]
26. Azam, E.S. Online damage detection in structural systems: Applications of proper orthogonal decomposition, and Kalman and particle filters. *Springerbriefs Appl. Sci. Technol.* **2014**, 1–45. [CrossRef]
27. Mirzazadeh, R.; Azam, E.S.; Mariani, S. Micromechanical characterization of polysilicon films through on-chip tests. *Sensors* **2016**, *16*. [CrossRef] [PubMed]
28. Cacace, F.; Germanib, A.; Palumboc, P. The observer follower filter: A new approach to nonlinear suboptimal filtering. *Automatica* **2013**, *49*, 548–553. [CrossRef]
29. Hao, G.; Sun, S.L.; Li, Y. Nonlinear weighted measurement fusion Unscented Kalman Filter with asymptotic optimality. *Inf. Sci.* **2015**, *299*, 85–98. [CrossRef]
30. Pomorski, K. Gauss-Hermite approximation formula. *Comput. Phys. Commun.* **2006**, *174*, 181–186. [CrossRef]
31. Santhanam, B.; Santhanam, T.S. On discrete Gauss-Hermite functions and eigenvectors of the discrete Fourier transform. *Signal Proc.* **2008**, *88*, 2738–2746. [CrossRef]
32. Strutinsky, V.M. Shells in deformed nuclei. *Nucl. Phys. A* **1968**, *122*, 1–33. [CrossRef]
33. Strutinsky, V.M. Shell effects in nuclear masses and deformation energies. *Nucl. Phys. A* **1967**, *95*, 420–442. [CrossRef]
34. Ge, Q.; Shao, T.; Chen, S.D.; Wen, C.L. Carrier Tracking Estimation Analysis by Using the Extended Strong Tracking Filtering. *IEEE Trans. Ind. Electron.* **2017**, *64*, 1415–1424. [CrossRef]
35. Moradkhani, H.; Hsu, K.L.; Gupta, H.; Sorooshian, S. Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resour. Res.* **2005**, *41*, 237–246. [CrossRef]
36. Yuen, K.V.; Hoi, K.I.; Mok, K.M. Selection of noise parameters for Kalman filter. *Earthq. Eng. Eng. Vib.* **2007**, *6*, 49–56. [CrossRef]
37. Kontoroupi, T.; Smyth, A.W. *Online Noise Identification for Joint State and Parameter Estimation of Nonlinear Systems*; ASCE: Reston, VA, USA, 2015.

38. Kitagawa, G. Non-Gaussian state space modeling of time series. In Proceedings of the 26th Conference on Decision and Control, Los Angeles, CA, USA, 9–11 December 1987; pp. 1700–1705.
39. Cong, J.; Li, Y.; Qi, G.; Sheng, A. An order insensitive sequential fast covariance intersection fusion algorithm. *Inf. Sci.* **2016**, *367–368*, 28–40. [CrossRef]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Cooperative Localization for Multi-AUVs Based on GM-PHD Filters and Information Entropy Theory

Lichuan Zhang, Tonghao Wang *, Feihu Zhang * and Demin Xu

School of Marine Science and Technology, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an 710072, China; zlc@nwpu.edu.cn (L.Z.); xudm@nwpu.edu.cn (D.X.)

* Correspondence: wangth@mail.nwpu.edu.cn (T.W.); feihu.zhang@nwpu.edu.cn (F.Z.)

Received: 7 August 2017; Accepted: 3 October 2017; Published: 8 October 2017

Abstract: Cooperative localization (CL) is considered a promising method for underwater localization with respect to multiple autonomous underwater vehicles (multi-AUVs). In this paper, we proposed a CL algorithm based on information entropy theory and the probability hypothesis density (PHD) filter, aiming to enhance the global localization accuracy of the follower. In the proposed framework, the follower carries lower cost navigation systems, whereas the leaders carry better ones. Meanwhile, the leaders acquire the followers' observations, including both measurements and clutter. Then, the PHD filters are utilized on the leaders and the results are communicated to the followers. The followers then perform weighted summation based on all received messages and obtain a final positioning result. Based on the information entropy theory and the PHD filter, the follower is able to acquire a precise knowledge of its position.

Keywords: cooperative localization (CL); multiple autonomous underwater vehicles (multi-AUVs); information entropy; probability hypothesis density (PHD) filter

1. Introduction

Localization is a primary issue for guaranteeing successful and efficient mission execution for autonomous underwater vehicles (AUVs) and other maritime robots. Common methods based on inertial measurement units (IMUs) have irreplaceable merit with respect to independencies, however, the accumulated error prevents high-accuracy localization in large-scale environments. In contrast to the use of more advanced IMUs, incorporating external information is a feasible solution. The authors of [1] considered the geomagnetic field sensor as an external information source during localization. Acoustic sensors such as sonars are also utilized in AUVs. Long baseline (LBL) [2,3] and ultra-short baseline (USBL) [4] systems often use sonars as an extension solution to support the localization. Sonar can also be used in underwater scenes of simultaneous localization and mapping (SLAM) [5]. The Doppler Velocity Log (DVL) is also a fundamental facility widely used for AUV navigation. An inertial navigation system (INV), DVL, and range measurement-based localization algorithm was designed for the AUV fleet [6]. Utilizing the DVL as an auxiliary sensor to INS, the authors of [7] exploited Unscented Kalman Filter (UKF) and Extended Kalman Filter (EKF) to locate the AUV positions. However, conventional methods work only in known areas (where essential facilities are deployed to offer standard reference), and are often restricted by the position of the sensors. Once the AUV flies beyond the border of these areas, the IMU system is the only option in the underwater environment. Cooperative localization (CL) is thus designed to address such problems by deploying sensors on the AUV. Moreover, CL can also extend the detection range to infinite regions, and enhance the reliability of the acoustic network.

According to the configuration of the navigation systems, CL is divided into parallel and the leader-follower structures. In the former structure, all AUVs carry same sensors, whereas in the latter,

the follower carries lower accuracy navigation systems. Normally, systems in parallel structures are more expensive for implementation. Roumeliotis [8] performed some related work with Kalman filter-based multi robot cooperative localization strategies. The authors of [9] compared the algebraic and EKF methods of AUV cooperative localization. An extended information filter was employed to address CL issues in [10]. Uncertainty and communication constraints were also considered in [11–13]. The authors of [14] proposed an optimization approach to jointly localize the multi-AUVs. However, the clutter is often caused by the echo from the air–water surface, seabed or some other objects, and not considered in AUV CL applications. Traditional approaches often utilize the data association to address the clutter, which is invalid once the surveillance region is small. The nearest neighbor (NN) association is a basic approach. The authors of [15] investigated the application of the NN association in scenarios of abrupt motion tracking. Research on global NN and suboptimal NN was reported in [16], which provided approaches for multiple target tracking. The joint probabilistic data association (JPDA) method is also a solution to the data association issue. However, its accuracy decreases once the targets get to a small region [17]. Another widely utilized data association approach is the multiple-hypothesis tracking (MHT) method [18]. In [19], the MHT method was utilized to track the targets in cluttered images. Without data association steps, the probability hypothesis density (PHD) filter is a novel method to solve multiple target tracking problem in cluttered environment. Based on the PHD filter, it is possible to solve the disturbance caused by the clutter. In the PHD filter, all states and measures are modeled as set-based values in the format of a random finite set (RFS), which makes the PHD filter a promising approach to solving these issues. The PHD filter has been used in CL for vehicles [20,21] and the experiment implied its effectiveness in clutter environments. The PHD filter was also introduced to address the visual tracking issue [22,23], aiming to solve the number variation and noise corruption of the camera. In this paper, we apply the PHD filter to solve the AUV cooperative localization problem. Meanwhile, information entropy theory is a proper tool for determining the quality of the estimations. It has been used in optimization [24] and industrial areas to value how useful one message is [25–27]. Therefore, in this paper, underwater clutter is considered as a disturbance and the PHD filter incorporated with information entropy theory is simultaneously employed.

The remainder of this paper is organized as follows. Section 2 introduces the background of this paper and the mathematic models of the AUV and some sensors. Some basic knowledge about the information entropy theory is also included. In Section 3, the problem is stated and the detailed steps of the designed algorithm are provided. Section 4 provides the simulation results in two cases, both of which support the validity of the proposed method. Finally, Section 5 gives the conclusions and points out the prospects of this paper.

2. Background

2.1. Assumptions

As shown in Figure 1:

1. The leader–follower structure is taken in order to achieve high accuracy with lower cost, in contrast to the parallel structure. Notice that the number of the leaders could be multiple, whereas the number of the followers is strictly limited to one.
2. Each AUV receives relative measurements of the others in the format of the range and bearing in local (body) coordinates. The global position of the follower can be calculated according to the measurement and the accurate position of the leader. However, the sonar cannot classify the clutter and the true measurements.
3. In this paper, only the communication between the leaders and the to-be-localized followers is considered. The measuring noise and connectivity uncertainty of the network are incorporated, whereas the time delay is beyond the scope of this paper.

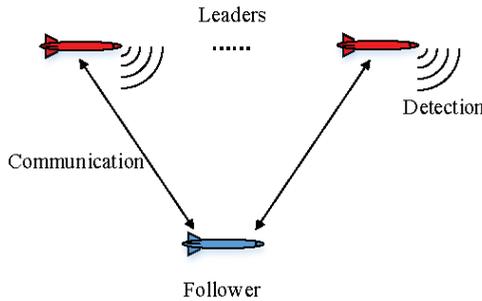


Figure 1. The leader–follower structure.

2.2. Mathematic Models

2.2.1. Model of AUV

As shown in Figure 2, the AUV is modeled in the two-dimensional horizontal plane, neglecting the roll motion. According to [28,29], the kinematic model is utilized to describe the relationship between the AUV body-fixed reference frame ($\{B\}$ -frame) and the Earth-fixed inertial reference frame ($\{I\}$ -frame). Note that the $\{I\}$ -frame is a simplified north–east–down (NED) system in the 2D horizontal plane, in which the X axis is pointing towards the north and the Y axis is pointing towards the east. The state vector of the AUV is then described as $X = [x \ y]^T$, describing the position of the origin of the $\{B\}$ -frame with respect to the $\{I\}$ -frame (i.e., the position of the AUV in relation to $\{I\}$). ψ is the orientation of the $\{B\}$ -frame with respect to $\{I\}$, i.e., the yaw of the AUV. $v = [u \ v]^T$ is the input of the AUV, denoting the linear velocity where u and v are the surge and sway speeds, respectively. r is the yaw rate of the AUV, i.e., $r = \dot{\psi}$.

Then the kinematic model of the AUV can be written as:

$$\begin{aligned} \dot{x} &= u \cos(\psi) - v \sin(\psi) \\ \dot{y} &= u \sin(\psi) + v \cos(\psi) \\ \dot{\psi} &= r \end{aligned} \tag{1}$$

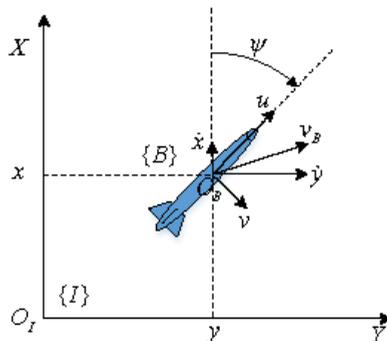


Figure 2. Two-dimensional model of the Autonomous Underwater Vehicle (AUV).

In the undersea environment, both the currents and the unexpected waves can interfere the movement of the AUV, and such noise is modeled as the additive systematic zero-mean white Gaussian noise vector $w_X(k) \in \mathbb{R}^2$ at instant t_k . We have

$$Q_X = E [w_X(k)w_X^T(k)] = \begin{bmatrix} \sigma_x^2 & \\ & \sigma_y^2 \end{bmatrix} \tag{2}$$

where Q_X represents the covariance matrix of w_X .

Similarly, the noise of the yaw ψ is modeled as $w_\psi(k) \in \mathbb{R}$, and the covariance is given by

$$Q_\psi = E [w_\psi(k)w_\psi^T(k)] = \sigma_\psi^2 \tag{3}$$

Thus, the real state (position) and yaw of the AUV can be written as:

$$\begin{aligned} X_{real} &= X_{ideal} + w_X \\ \psi_{real} &= \psi_{ideal} + w_\psi \end{aligned} \tag{4}$$

Only X_{real} is used in the remainder of this paper, so it will be written as just X for clarity. Notice that since the leaders are assumed to carry high performance sensors, and the observations are carried out by the leaders, the yaw sensing error is neglected.

2.2.2. Model of the Detecting Sonar

The AUV often relies on acoustic sonars to observe the surrounding environments. Notice that sonar signal processing includes transmitting, flying and receiving, etc.

A. Relative Distance-Sensing Part

The two-way time-of-flight sensor is utilized to compute the relative distance. Assuming that the time period between transmitting and receiving the signal is t_f and the underwater sound velocity is C , the relative distance $Dist$ can be expressed as

$$Dist = t_f \cdot C \tag{5}$$

Figure 3 shows that sonar i detects the AUV j at instant k . The distance measurement $z_{ij}^d(k)$ is then calculated by

$$z_{ij}^d(k) = \|X_j - X_i\| + v_p(k) = \sqrt{(x_j(k) - x_i(k))^2 + (y_j(k) - y_i(k))^2} + v_p(k) \tag{6}$$

where X_i and X_j are the states of the AUV i and the sonar j . $v_p(k)$ is the noise and is modeled by the additive stochastic zero-mean white Gaussian noise with variance $R_p(k) = \sigma_p^2$.

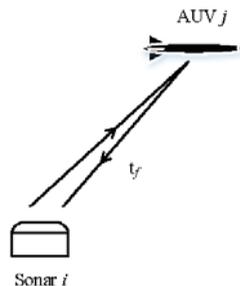


Figure 3. Schematic diagram of the relative distance detection.

Remark 1. Here we define the state of a sonar in the same way as the AUV for convenience. Normally, the sonar is fixed on the AUV in the head part sharing the forward direction with respect to the AUV. The third term of the vector X_j , denoting the yaw of the AUV, plays an important role in direction sensing.

B. Relative Direction-Sensing Part

The sonar is usually formed by a hydrophone array. Take Figure 4 as an example, A and B are two hydrophones in a sonar, and AUV *i* is the target to be detected. Signals from different direction reflect different hydrophones in a specific order. Hence, signal reflected by AUV *i* arrives at hydrophone B earlier than A.

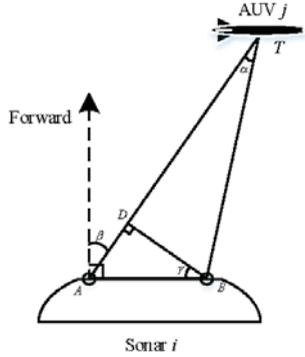


Figure 4. Schematic diagram of the relative direction detection.

Therefore, the time period τ_{AB} between the arrival instants can be used to compute the relative direction of the AUV against the sonar. Generally, the distance between the AUV and the detecting sonar (*TA* or *TB*) is much longer than *AB*. Assuming $\cos \alpha = 1$, and $AD = AB$, we have

$$AD = \tau_{AB} \cdot C \tag{7}$$

Thus, the relative direction is given by

$$\beta = \arcsin(\tau_{AB} \cdot C / AB) + v_{\beta}(k) \tag{8}$$

where $v_{\beta}(k)$ is the stochastic zero-mean white Gaussian noise with variance $R_d(k) = \sigma_d^2$. $z_{ij}^r(k)$ denotes the relative direction measure between the sensor *j* and the target *j* at t_k .

Combing the relative distance and the direction (i.e. the relative measurements z_{ij}^r), the position of the AUV is acquired (see Figure 5).

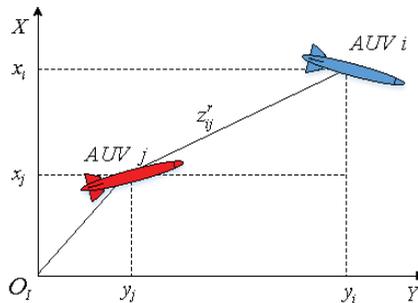


Figure 5. Position calculation.

2.3. PHD Filter

As aforementioned, observations from sonars also contain clutter due to the complicated underwater environment. Therefore, the standard Bayesian filter cannot obtain satisfactory estimations without data association. Compared to the standard Bayesian filters, finite set statistics analysis is a convenient approach for achieving the task without considering data association [30,31]. Hence, the RFS-based Gaussian mixture-probability hypothesis density (GM-PHD) filter is utilized.

Suppose that there are N_{k-1} targets at instant t_{k-1} with states $X_{k-1} = \{x_1(k-1), x_2(k-1), \dots, x_{N_{k-1}}(k-1)\}$, and there are N_k targets $X_k = \{x_1(k), x_2(k), \dots, x_{N_k}(k)\}$ including newborn and dead ones. With the same manner, both Z_{k-1} and Z_k are defined. In an RFS manner, we have

$$\begin{aligned} X_k &= \{x_1(k), x_2(k), \dots, x_{N_k}(k)\} \in F(X) \\ Z_k &= \{z_1(k), z_2(k), \dots, z_{N_k}(k)\} \in F(Z) \end{aligned} \tag{9}$$

where $F(X)$ and $F(Z)$ denote the sets containing all states and observations, respectively. Notice that Z contains both the real measure of the target and the clutter.

Then, Equation (9) can be simplified as

$$\begin{aligned} X_k &= \left(\bigcup_{\zeta \in X(k-1)} S_{k|k-1}(x) \right) \cup \Gamma_k \\ Z_k &= \left(\bigcup_{x \in X(k)} \Theta_k(x) \right) \cup K_k \end{aligned} \tag{10}$$

where $S_{k|k-1}$ is the RFS of remaining targets at t_{k-1} , Γ_k is the set of newborn ones at t_k , $\Theta_k(x)$ is the measure set of the targets, and K_k is the clutters RFS.

Given the survival and detection rate as $p_{s,k}$ and $p_{d,k}$, the GM-PHD filter can be expressed in a Bayesian prediction-and-update form as

Prediction:

$$\begin{aligned} v_{k|k-1}(x) &= v_{s,k|k-1}(x) + \gamma_k(x) \\ v_{s,k|k-1}(x) &= p_{s,k} \sum_{j=1}^{J_{k-1}} \omega_{k-1}^j N(x; m_{s,k|k-1}^j, P_{s,k|k-1}^j) \\ m_{s,k|k-1}^j &= F_{k-1} m_{k-1}^j \\ P_{s,k|k-1}^j &= Q_{k-1} + F_{k-1} P_{k-1}^j F_{k-1}^T \end{aligned} \tag{11}$$

Update:

$$\begin{aligned}
 v_k(x) &= (1 - p_{d,k})v_{k|k-1}(x) + \sum_{z \in Z_k} v_{d,k}(x; z) \\
 v_{k|k-1}(x) &= \sum_{i=1}^{J_{k|k-1}} \omega_{k-1}^i N(x; m_{k|k-1}^i, P_{k|k-1}^i) \\
 v_{d,k}(x; z) &= \sum_{j=1}^{J_{k|k-1}} \omega_{k-1}^j N(x; m_{k|k}^j, P_{k|k}^j) \\
 \omega_{k-1}^i &= \frac{p_{d,k} \omega_{k-1}^i q_k^i(z)}{\kappa_k(z) + p_{d,k} \sum_{l=1}^{J_{k|k-1}} \omega_{k|k-1}^l q_k^l(z)} \\
 q_k^i(z) &= N(z; H_k m_{k|k}^i, R_k + H_k P_{k|k-1}^i H_k^T) \\
 m_{k|k}^i(z) &= m_{k|k-1}^i + K_k^i (z - H_k m_{k|k-1}^i) \\
 P_{k|k}^i &= [I - K_k^i H_k] P_{k|k-1}^i \\
 K_k^i &= P_{k|k-1}^i H_k^T (H_k P_{k|k-1}^i H_k^T + R_k)^{-1}
 \end{aligned} \tag{12}$$

where F_k and H_k are the transition matrix and the measurement matrix, respectively. Note that if the transition function f_k and the measurement function h_k are nonlinearly presented, we can take their Jacobian matrices as F_k and H_k . γ_k denotes the Gaussian mixture and is given by

$$\gamma_k(x) = \sum_{i=1}^{J_{\Gamma,k}} \omega_{\Gamma,k}^i N(x; m_{\Gamma,k}^i, P_{\Gamma,k}^i) \tag{13}$$

Although the PHD filter estimates both the statements and its number, it does not have all of the track information. Thus, the identification of a specific state in its life cycle is not available. Hence, we have the information entropy theory to address this issue.

2.4. Information Entropy Theory

Normally, the standard PHD filter gives several results, which can fluctuate around the true states. However, in this paper, the follower needs only one estimation of its position. Thus, a proper method to calculate only one estimation with respect to all the results given by standard PHD filter is needed. Information entropy is one feasible tool to evaluate the quality or usefulness of the estimation. Therefore, it is employed to derive the weight coefficients for the PHD estimations.

Given a certain event a_i with probability $p(a_i)$, the amount of information brought by a_i can be given by

$$I(a_i) = \log_b \frac{1}{p(a_i)} = -\log_b p(a_i) \tag{14}$$

where $I(a_i)$ denotes the self-information and b is the base value of logarithm. Shannon defined the expected value of $I(a_i)$ as information entropy, given by

$$\begin{aligned}
 H(X) &= E [I(X)] = \sum_{i=1}^n p(x_i) I(x) \\
 &= - \sum_{i=1}^n p(x_i) \log_b p(x_i)
 \end{aligned} \tag{15}$$

where E is the expect operator. Thus, information entropy can be a metric with respect to the uncertainty of an information source.

When considering several independent events (here we take two for example), the joint entropy and the conditional entropy can be written as

$$H(X, Y) = - \sum_i \sum_j p(x_i, y_j) \log p(x_i, y_j) \tag{16}$$

$$\begin{aligned} H(X|Y) &= \sum_i \sum_j p(x_i, y_j) \log \frac{p(y_j)}{p(x_i, y_j)} \\ &= - \sum_i \sum_j p(y_j) p(x_i|y_j) \log p(x_i|y_j) \end{aligned} \tag{17}$$

where X and Y denote two independent events with respect to x_i and y_i , respectively.

Since $H(X)$ and $H(X|Y)$ describe the prior and posterior uncertainties, their difference is defined as mutual information as

$$I(X; Y) = H(X) - H(X|Y) \tag{18}$$

When X is defined as n -dimensional Gaussian distribution, i.e., $X \sim N(\mu_Y, P_Y)$, and taking e as the base to simplify the computation, the entropy can be written as

$$H(X) = 0.5 \ln(2\pi e)^n |P_X| \tag{19}$$

3. Algorithm Design

In scenarios of N leader AUVs and one follower AUV j , all the AUVs can accurately obtain their positions on the sea surface. Once both the leaders and the follower dive into the water, the INS is utilized to localize them. The leaders can obtain accurate positions with the help of highly accurate navigation sensors, in contrast to the follower. To enhance the positioning accuracy of the follower, the leaders should also estimate the position of the follower via relative measures. However, due to the complicated underwater environment, the clutter issue is quite challenging. The GM-PHD filter is thus utilized to filter the clutter from the whole measurement set.

However, all the leaders give different estimations with respect to the same follower. For better execution, the follower needs only one accurate position of itself. In this paper, a specific algorithm is designed for the follower to cope with all the estimations from different sources properly. The process of the algorithm is as follows.

Step 1: Measuring. At t_k , the leaders measure the follower. For instance, the leader AUV i collects the observations z_{k-1}^{ij} including the true measurements of the follower AUV j z_{k-1}^{ij} and the clutter c_k^{ij} , i.e., $z_{k-1}^{ij} = \{z_{k-1}^{ij}\} \cup \{c_{k-1}^{ij}\}$.

Step 2: Calculating. With the help of the GM-PHD filter, each leader computes the estimated positions of the follower in the format of $\hat{X}_{k|k}^{ij,t}$. Moreover, according to Equations (17) and (19), the entropy of $\hat{X}_{k|k-1}^{ij,t}$ and $\hat{X}_{k|k}^{ij,t}$ are also calculated by the leaders, denoted as

$$\begin{aligned} H(\hat{X}_{k|k-1}^{ij,t}) &= 0.5 \ln(2\pi e)^n |P_{k|k-1}| \\ H(\hat{X}_{k|k}^{ij,t}) &= 0.5 \ln(2\pi e)^n |P_{k|k}| \end{aligned} \tag{20}$$

where $P_{k|k-1}$ and $P_{k|k}$ are the prior and posterior covariance matrices, respectively. Thus, the mutual information is given by

$$I_k^{ij,t} = I(\hat{X}_{k|k-1}^{ij,t}; \hat{X}_{k|k}^{ij,t}) = H(\hat{X}_{k|k-1}^{ij,t}) - H(\hat{X}_{k|k}^{ij,t}) \tag{21}$$

which describes the amount of uncertainty eliminated by $\hat{X}_{k|k}^{ij,t}$.

Step 3: Communicating. Leaders broadcast both the estimated positions and the corresponding mutual information values.

Step 4: Fusing. The follower AUV j collects all the estimations and mutual information values from the leaders. Note that the number of the data pairs is usually larger than the number of the leaders due to the property of the PHD filter. The more uncertainty is eliminated, the better the estimation. Assuming that the leader AUV i gives M_i pairs of the results, the weighted summation of all the estimations is given by

$$\hat{X}_{k|k}^j = \sum_{i=1}^N \sum_{t=1}^{M_i} w_{it} \hat{X}_{k|k}^{ij,t} \tag{22}$$

where w_{it} is the normalized weighting coefficient computed by

$$w_{it} = \frac{I(\hat{X}_{k|k-1}^{ij,t}; \hat{X}_{k|k}^{ij,t})}{\sum_{i=1}^N \sum_{t=1}^{M_i} I(\hat{X}_{k|k-1}^{ij,t}; \hat{X}_{k|k}^{ij,t})} \tag{23}$$

Thus, the calculated $\hat{X}_{k|k}^j$ is the final estimation of the follower AUV j , which is then broadcasted by the follower to the leaders, for further PHD filtering steps.

Step 5: Back to Step 1.

Figure 6 presents a brief procedure of the proposed algorithm.

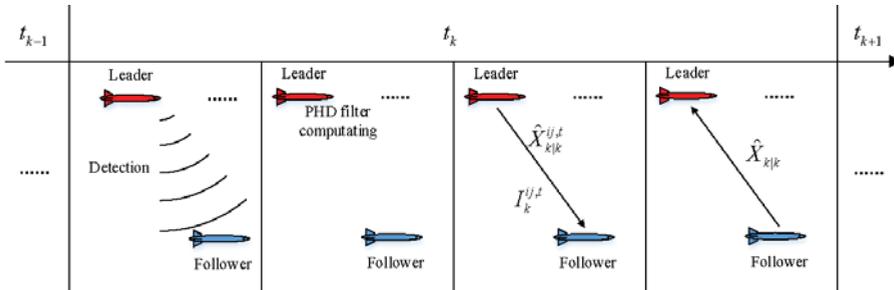


Figure 6. Working procedure of the proposed algorithm. PHD: Probability Hypothesis Density.

4. Simulation and Results

Simulations are designed to validate the proposed method, under the assumed communication conditions with noise and intermittent network failure. Two scenarios are designed as follows.

Scenario 1:

There were two leading AUVs in this scenario, AUV 1 and AUV 2, and they both took straight flights with a speed of 2 kn. Starting from the origin of the Earth-fixed frame, their directions were northeast (45° to the north) and west, respectively. Their measuring frequency was set to 1 Hz. The follower AUV 3 took a circular motion on the 2D square surveillance region with the x -range $[-200 \text{ m}, 300 \text{ m}]$ and y -range $[-400 \text{ m}, 200 \text{ m}]$. AUV 3 started its flight at $(0, 0)$ with cruising speed of 5 kn and yaw angular velocity 0.05 rad/s. Note that the angular velocity is positive clockwise in the NED frame. The motion noise was assumed to be zero mean white Gaussian noise with covariance matrix $Q = \text{diag}([1 \text{ m}^2, 1 \text{ m}^2, 10^{-4} \text{ rad}^2])$, while the covariance matrix of the measurement noise was $R = \text{diag}([25 \text{ m}^2, 25 \text{ m}^2])$. The clutter was uniformly distributed over the surveillance region and the number of the clutter observations followed $N_{Clutters} \sim P(\lambda_{Clutters})$, where $\lambda_{Clutters}$ is calculated by the area of the region and the occurrence probability $p_{Clutters}$. Here the $p_{Clutters} = 12.5 \times 10^{-6} / \text{m}^2$. The network failure ratio was set to 10% and the detection failure ratio was set to 2%.

The trajectories under this condition are shown in Figure 7.

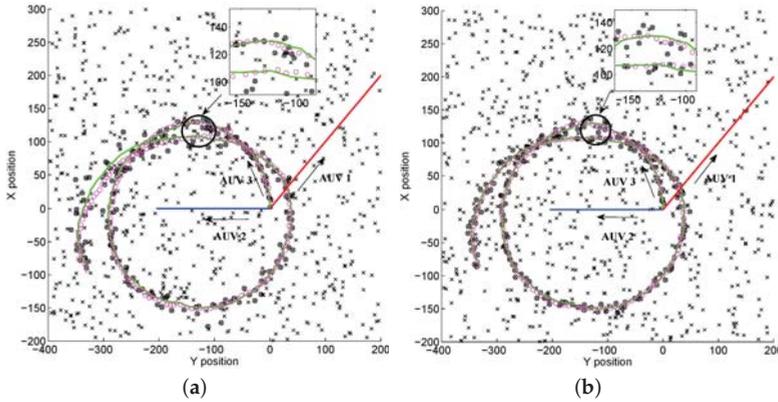


Figure 7. Trajectories, clutter, and measurements in Scenario 1. (a) Result with respect to AUV 1 only; (b) Result with respect to AUV 2 only.

The red, blue and green lines in Figure 7a,b are the actual trajectories of the leaders AUV 1 and 2, and the follower AUV 3, respectively. Black circles represent the real measures of AUV 3 and the "x" markers show the clutter. The magenta circles give the results of the standard PHD filter executed by one single leader.

As aforementioned, the number of the clutter observations is randomly generated according to the Poisson distribution. In this simulation, the number is plotted in Figure 8.

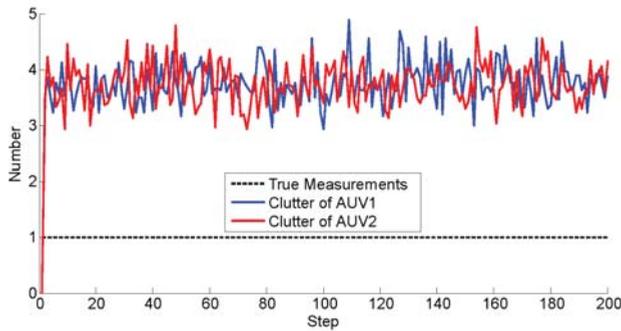


Figure 8. Number of the clutter observations.

We note that the true measures are merged in the clutter. Thus, the PHD filter was incorporated to eliminate the effects of the clutter during filtering. Figure 9 illustrates the number of the PHD estimations.

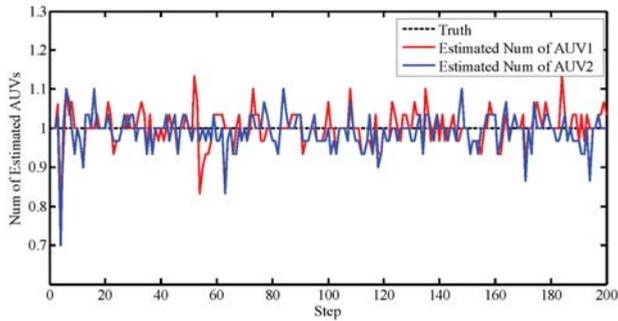


Figure 9. Number of estimations.

Notice that without data association, the estimated number fluctuates around the true value. From Figure 9, it is observed that the PHD cannot eliminate the clutter completely, but compared with Figure 8, the number has been adjusted to a small range.

To better evaluate the proposed approach, a metric was needed to justify the accuracies. Notice that the standard Root-Mean-Square Error (RMSE) cannot be directly used due to the random number of the set-valued states. Therefore, the Optimal Sub-Pattern Assignment (OSPA) was utilized to describe the accuracy of the localization. OSPA was first proposed in [32] as a metric in the multi-object filtering field, describing the miss-distance, or error, between the reference values and the estimated ones. Note that using OSPA, the number of reference values and the estimations do not have to be the same. In other words, OSPA offers a metric between two sets regardless the cardinalities. A larger value, i.e., larger miss-distance, indicates worse accuracy. Here we plotted the OSPA distance lines of different algorithms in Figure 10. Note that the OSPA error lines were obtained after 100 Monte Carlo simulations, and the saturation value of the OSPA is set to 50.

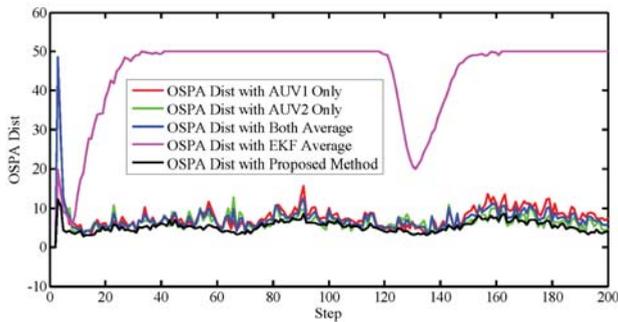


Figure 10. Positioning error. OSPA: Optimal Sub-Pattern Assignment, Dist: Distance, EKF: Extended Kalman Filter.

The EKF with the nearest neighbor rule was also carried out as the comparison to the PHD filter. Generally, the EKF performs the poorest, implying the superiority of the PHD filter. Also, the OSPA distance of the proposed method is smaller than both the methods exploiting one AUV and the alternative solutions using them both, which illustrates the advantage of the proposed method.

Scenario 2:

In this scenario, the AUVs cruise in different manner. The cruise velocities were 5 kn for the leaders AUV 1 and AUV 2, and 2 kn for the follower AUV 3. As shown in Figure 11, the AUV 1 started

from (0, -250) in the west of the origin of the NED frame, to the east in a sine wave routine, while the AUV 2 took a circular path from (30, 30), with yaw rate 0.05 rad/s. AUV 3 traveled on the horizontal region with both the x -range and y -range $[-300,300]$, from the northwest (-250, 250) straight to the southeast in 3 kn. Compared to Scenario 1, the network failure rate increased to 20% and the motion (systematic) noise of the leaders was also considered. The rest parameters were set up in the same manner as for Scenario 1.

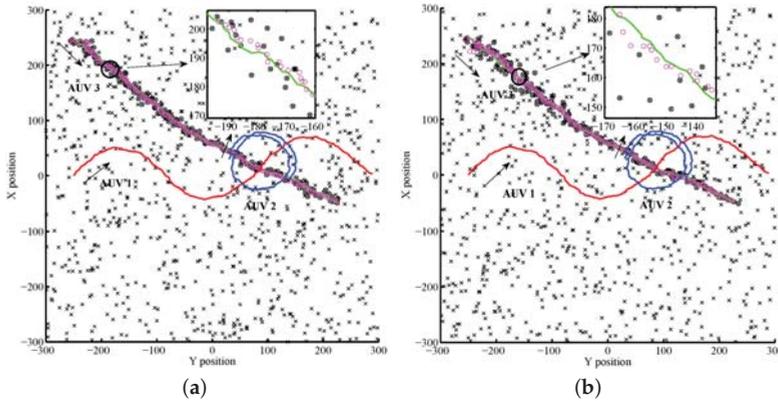


Figure 11. Trajectories, clutter and measurements in Scenario 2. (a) Result with respect to AUV 1 only; (b) Result with respect to AUV 2 only.

The trajectories are described as red line (leader AUV 1), blue line (leader AUV 2), and green line (follower AUV 3) in Figure 11a,b. The real measures of AUV 3 are denoted as black circles and the "x" markers represent the clutter. The results of the standard PHD filter executed by one single leader are given by the magenta.

The number of the clutter observations and PHD estimations are given in Figures 12 and 13.

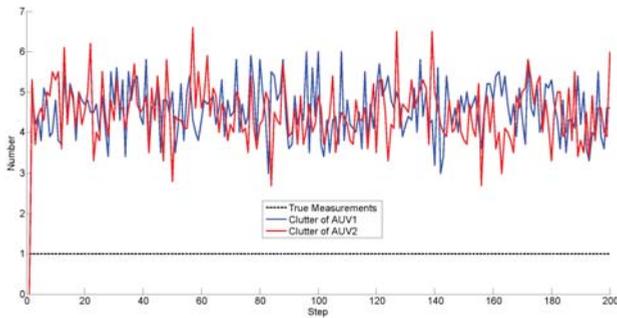


Figure 12. Number of the clutter observations.

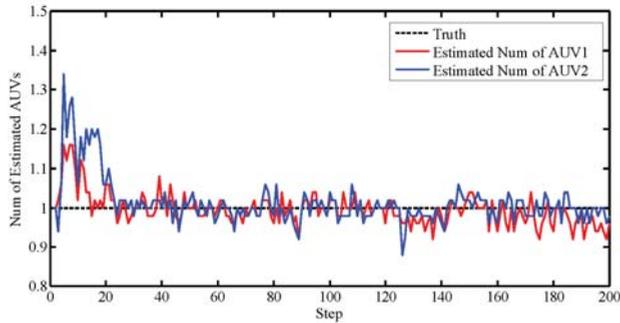


Figure 13. Number of estimations.

Figure 12 and Figure 13 illustrate that the PHD filter still managed to reduce the influence of the clutter. The OSPA distances with respect to different algorithms are shown in Figure 14.

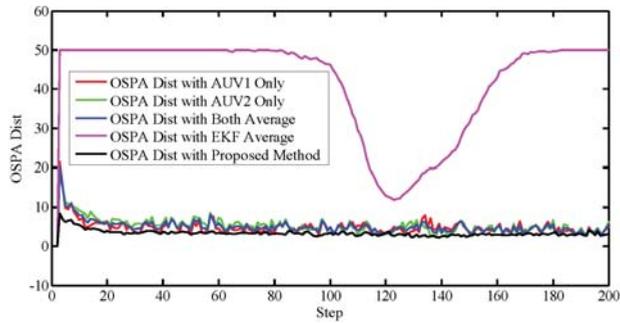


Figure 14. Positioning error.

In this scenario, the proposed method still achieved the best performance compared to the EKF, the single PHD filter, and the average of them.

However, the computational load slightly increases due to the calculation of mutual information. Running on a computer with Intel(R) Core(TM) i7-4710MQ CPU at 2.5GHz, the computing time of 50 executions of the standard PHD filter and the proposed method in different scenarios are given in the Table 1.

Table 1. Computing time of the standard PHD filter and the proposed method.

	Standard PHD Filter (s)	Proposed Method (s)	Increased Percentage (%)
Scenario 1	211.602	229.887	8.64
Scenario 2	247.366	267.481	8.13

According to the simulation results, the PHD filter and information entropy-based cooperative localization algorithm achieved higher accuracy and stability compared to the state-of-the-art approaches. The computation burden is increased by computing the mutual information, but according to the Table 1, the loads were only increased by around 8%.

5. Conclusions

In this paper, an information entropy-based GM-PHD filter was employed to tackle the cooperative localization issue. By exploiting both the GM-PHD filter and the information entropy theory, the proposed method can obtain more accurate results in contrast to the standard GM-PHD filter and the EKF approach with NN association strategy. Mathematical models were derived and a simulation was designed, which implied that the designed method performed well in scenarios of clutter, intermittent connection, and detection failure.

There are also some prospects that need to be investigated in future. One is to explore the performance of the proposed method in the communication environment with time delay. The time delay issue results in a bottleneck of underwater CL, which can decrease the positioning accuracy. A deeper insight into this is required. Also, observability is an important feature of the multi-AUV system [33]. The influence of the observability in the proposed method should be investigated. Moreover, how to utilize the proposed method in scenarios of multiple followers is the key to applying this algorithm more widely.

Besides the theoretical improvements of the proposed algorithm, experimental validations should also be executed in the future for a more convincing conclusion. Primary experiments can be implemented in artificial environments like swimming pools with preset clutter, or lakes, and the AUVs can be replaced by sonar and acoustic communication units for simplification. Then, several AUVs or some other underwater robots with similar motion patterns modeled in this paper should be exploited and a sea experiment will be needed to validate the proposed method in practice.

Acknowledgments: This work was supported by the Northwestern Polytechnical University (NWPU).

Author Contributions: Lichuan Zhang and Tonghao Wang performed the simulations and wrote the paper; Feihu Zhang revised this paper and offered some useful suggestions in methodologies and simulations; Demin Xu offered revision during the whole process.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Huang, Y.; Hao, Y. Method of separating dipole magnetic anomaly from geomagnetic field and application in underwater vehicle localization. In Proceedings of the 2010 IEEE International Conference on Information and Automation, Harbin, China, 20–23 June 2010; pp. 1357–1362.
- Matos, A.; Cruz, N.; Martins, A.; Lobo Pereira, F. Development and implementation of a low-cost LBL navigation system for an AUV. In Proceedings of the OCEANS '99 MTS/IEEE, Seattle, WA, USA, 13–16 September 1999; Volume 2, pp. 774–779.
- Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV Navigation and Localization: A Review. *IEEE J. Ocean. Eng.* **2014**, *39*, 131–149.
- Allotta, B.; Bartolini, F.; Caiti, A.; Costanzi, R.; Di Corato, F.; Fenucci, D.; Gelli, J.; Guerrini, P.; Monni, N.; Munafò, A.; et al. Typhoon at CommsNet13: Experimental experience on AUV navigation and localization. *Annu. Rev. Control* **2015**, *40*, 157–171.
- Stachniss, C.; Leonard, J.J.; Thrun, S. Simultaneous Localization and Mapping. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 1153–1176.
- Allotta, B.; Pugi, L.; Costanzi, R.; Vettori, G. Localization algorithm for a fleet of three AUVs by INS, DVL and range measurements. In Proceedings of the 2011 15th International Conference on Advanced Robotics (ICAR), Tallinn, Estonia, 20–23 June 2011; pp. 631–636.
- Karimi, M.; Bozorg, M.; Khayatian, A. A comparison of DVL/INS fusion by UKF and EKF to localize an autonomous underwater vehicle. In Proceedings of the 2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 13–15 February 2013; pp. 62–67.
- Huang, G.P.; Trawny, N.; Mourikis, A.I.; Roumeliotis, S.I. Observability-based consistent EKF estimators for multi-robot cooperative localization. *Auton. Robots* **2011**, *30*, 99–122.

9. Zhang, L.; Wang, J.; Wang, T.; Liu, M.; Gao, J. Optimal formation of multiple AUVs cooperative localization based on virtual structure. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016; pp. 1–6.
10. Rui, G.; Chitre, M. Cooperative multi-AUV localization using distributed extended information filter. In Proceedings of the 2016 IEEE/OES Autonomous Underwater Vehicles (AUV), Tokyo, Japan, 6–9 November 2016; pp. 206–212.
11. Nerurkar, E.D.; Roumeliotis, S.I. Asynchronous Multi-Centralized Cooperative Localization. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010.
12. Nerurkar, E.D.; Roumeliotis, S.I. A communication-bandwidth-aware hybrid estimation framework for multi-robot cooperative localization. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 1418–1425.
13. Nerurkar, E.D.; Zhou, K.X.; Roumeliotis, S.I. A hybrid estimation framework for cooperative localization under communication constraints. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 502–509.
14. Ferreira, B.Q.; Gomes, J.; Soares, C.; Costeira, J.P. Collaborative localization of vehicle formations based on ranges and bearings. In Proceedings of the 2016 IEEE Third Underwater Communications and Networking Conference (UComms), Lercici, Italy, 30 August–1 September 2016; pp. 1–5.
15. Zhou, T.; Lu, Y.; Lv, F.; Di, H.; Zhao, Q.; Zhang, J. Abrupt motion tracking via nearest neighbor field driven stochastic sampling. *Neurocomputing* **2015**, *165*, 350–360.
16. Konstantinova, P.; Udvarov, A.; Semerdjiev, T. A study of a target tracking algorithm using global nearest neighbor approach. In Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech '03), Rousse, Bulgaria, 19–20 June 2003; pp. 290–295.
17. Fitzgerald, R. Track biases and coalescence with probabilistic data association. *IEEE Trans. Aerosp. Electr. Syst.* **1985**, *AES-21*, pp. 822–825.
18. Blackman, S.S. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerosp. Electr. Syst. Mag.* **2004**, *19*, 5–18.
19. Chenouard, N.; Bloch, I.; Olivo-Marin, J.C. Multiple hypothesis tracking for cluttered biological image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2736–2750.
20. Zhang, F.; Buckl, C.; Knoll, A. Multiple Vehicle Cooperative Localization with Spatial Registration Based on a Probability Hypothesis Density Filter. *Sensors* **2014**, *14*, 995–1009.
21. Goli, A.S.; Far, H.B.; Fapojuwo, O.A. Cooperative Multi-sensor Multi-vehicle Localization in Vehicular Adhoc Networks. In Proceedings of the 2015 IEEE International Conference on Information Reuse and Integration, San Francisco, CA, USA, 13–15 August 2015; pp. 142–149.
22. Feng, P.; Wang, W.; Dlay, S.; Naqvi, S.M.; Chambers, J. Social Force Model-Based MCMC-OCSVM Particle PHD Filter for Multiple Human Tracking. *IEEE Trans. Multimedia* **2017**, *19*, 725–739.
23. Zhou, X.; Li, Y.; He, B.; Bai, T. GM-PHD-Based Multi-Target Visual Tracking Using Entropy Distribution and Game Theory. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1064–1076.
24. Zhou, X.; Zhang, Y.; Hao, S.; Li, S. A new approach for noise data detection based on cluster and information entropy. In Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 8–12 June 2015; pp. 1416–1419.
25. Zou, G.; Ma, L.; Zhang, L.; Mo, L. An indoor positioning algorithm using joint information entropy based on WLAN fingerprint. In Proceedings of the 2014 International Conference on Computing, Communication and Networking Technologies (ICCCNT), Hefei, China, 11–13 July 2014; pp. 1–6.
26. Mocnej, J.; Lojka, T.; Zolotov, T. Using information entropy in smart sensors for decentralized data acquisition architecture. In Proceedings of the 2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMII), Herlany, Slovakia, 21–23 January 2016; pp. 47–50.
27. Guo, X.; Liu, X.; Chen, G.; Chang, L.; Li, X. Missile weapon system-of-systems optimization method based on information entropy. In Proceedings of the 2016 International Conference on Computer, Information and Telecommunication Systems (CITS), Kunming, China, 6–8 July 2016; pp. 1–5.
28. Fossen, T.I. *Guidance and Control of Ocean Vehicles*; John Wiley & Sons Inc: Chichester, UK; New York, NY, USA; Brisbane, Australia; Toronto, Canada; Singapore, 1994.

29. Lapiere, L.; Jouvencel, B. Robust nonlinear path-following control of an AUV. *IEEE J. Ocean. Eng.* **2008**, *33*, 89–102.
30. Mahler, R.P.S. “Statistics 101” for multisensor, multitarget data fusion. *IEEE Aerosp. Electr. Syst. Mag.* **2004**, *19*, 53–64.
31. Mahler, R. “Statistics 102” for Multisource-Multitarget Detection and Tracking. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 376–389.
32. Schuhmacher, D.; Vo, B.T.; Vo, B.N. A Consistent Metric for Performance Evaluation of Multi-Object Filters. *IEEE Trans. Signal Process.* **2008**, *56*, 3447 – 3457.
33. Arrichiello, F.; Antonelli, G.; Aguiar, A.P.; Pascoal, A. Observability metric for the relative localization of AUVs based on range and depth measurements: Theory and experiments. In Proceeding of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 3166–3171.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Research on the Fusion of Dependent Evidence Based on Rank Correlation Coefficient

Fengjian Shi, Xiaoyan Su *, Hong Qian, Ning Yang and Wenhua Han

School of Automation Engineering, Shanghai University of Electric Power, Shanghai 200090, China; shifengjian@mail.shiep.edu.cn (F.S.); qianhong@shiep.edu.cn (H.Q.); yangning@shiep.edu.cn (N.Y.); hanwenhua@shiep.edu.cn (W.H.)

* Correspondence: suxiaoyan@shiep.edu.cn; Tel.: +86-21-3530-3543

Received: 1 September 2017; Accepted: 9 October 2017; Published: 16 October 2017

Abstract: In order to meet the higher accuracy and system reliability requirements, the information fusion for multi-sensor systems is an increasing concern. Dempster–Shafer evidence theory (D–S theory) has been investigated for many applications in multi-sensor information fusion due to its flexibility in uncertainty modeling. However, classical evidence theory assumes that the evidence is independent of each other, which is often unrealistic. Ignoring the relationship between the evidence may lead to unreasonable fusion results, and even lead to wrong decisions. This assumption severely prevents D–S evidence theory from practical application and further development. In this paper, an innovative evidence fusion model to deal with dependent evidence based on rank correlation coefficient is proposed. The model first uses rank correlation coefficient to measure the dependence degree between different evidence. Then, total discount coefficient is obtained based on the dependence degree, which also considers the impact of the reliability of evidence. Finally, the discount evidence fusion model is presented. An example is illustrated to show the use and effectiveness of the proposed method.

Keywords: D–S evidence theory; dependent evidence; rank correlation coefficient

1. Introduction

With the development of science and technology, in order to meet the higher accuracy and system reliability requirements, the information fusion filters for multi-sensor systems have been widely applied [1]. Various methods have been proposed for multi-sensor modeling and sensor data fusion [2–7], including night-vision image fusion [8], weighted measurement fusion and Unscented Kalman Filter [9], neural network models [10], fuzzy set theory [11], belief function theory [12], and so on.

The fusion of different uncertain data is an important research topic of modern intelligent multi-sensor systems. Among these methods, Dempster–Shafer evidence theory (D–S theory) has been investigated for many applications in multi-sensor information fusion due to its flexibility in uncertainty modeling [13–15]. D–S theory was first proposed by Dempster in 1967 [16], and further developed by Shafer in 1976 [17]. It can not only deal with imprecise information and uncertain information, but also deal with complimentary information and missing information [18–21]. Therefore, besides multi-sensor information fusion, D–S theory has also been investigated for applications in many fields such as fault diagnosis [22,23], pattern recognition [24–28], multi-source information fusion [29], multiple attribute decision making [30–36] and risk analysis [37–40].

In D–S theory, Dempster’s rule plays a vital part in the process of information fusion. However, there is an issue that limits the application of evidence theory. The classical evidence theory is based on the assumption that the evidence is independent of each other [41]. In practice, the dependence is more common. In other words, some elementary item of evidence will be counted

twice without considering dependent evidence in the process of information fusion [42]. For example, one expert's opinion may be affected by another expert's opinion in an open decision making environment. In addition, there will be dependence between the evidence "grain production" and the evidence "natural disaster" in the agriculture risk analysis system [43]. A mistake will be generated if ignoring dependence between different evidence. To deal with dependent sources of information, many scholars proposed different methods [44]. The existing methods could be divided into two categories [45]: (1) improve the combination rule or (2) modify the original belief structure.

For the first category, the basic idea is to find a new evidence fusion method without considering dependence [45]. Some scholars have proposed their own combination rules. Cattaneo proposed the rules based on an assumption of minimal conflict rather than the independence assumption [46]. However, this method only analyzes the dependence between BBAs (Basic Belief Assignments) and fails to reveal the dependence between information sources. Destercke holds the view that the minimum rule of possibility theory could be generalized to the dependent evidence fusion [47,48]. However, in [18], the author thought this rule didn't satisfy the fundamental evidence equation. In [42], the cautious rule of combination aiming at reliable sources of evidence and the bold disjunctive rule aiming at unreliable sources of evidence is proposed. Both of them satisfy commutative law, associative law and idempotent law. However, this combination rule was established in the canonical decomposition of BBA. Choenni proposed a dependent evidence fusion method using the idea of joint probability distribution of probability theory [49]. However, this method essentially deals with BBA as a discrete probability distribution and the fusion result is a couple of focal elements rather than BBA. Chebbah et al. proposed a new combination rule that takes consideration of sources' degree of independence and they also suggest a method to quantify sources' degree of independence [50].

For the second category, the basic idea is to reduce the repetitive computation of the dependent part of the information sources as far as possible [45]. In this category, there are two research ideas. The first research idea is based on the relevant source evidence model, which is first proposed in [51] by Smets. This paper holds that the reason why the two pieces of evidence are related is that they are obtained from the same source of evidence. The same source of evidence represents the correlation part between the pieces of evidence. Smets proposed a combination method in [51] based on the TBM (transferable belief model). Then, Xiao et al. proposed a combination rule based on the model in [51], and this rule is in the framework of D-S evidence theory. According to this theory, if we know the correlation part of two information sources, and the evidence of the correlation information source is available, this method is effective to deal with dependent evidence. However, this method is not reasonable as it does not care about the significance of the common evidence in some application systems. In addition, how to acquire the common evidence between two dependent pieces of evidence remains a question.

The second research idea is based on the discount evidence model. The main idea of this model is that dependent evidence shouldn't be given the same weight as independent evidence in the process of information fusion since it provides less effective information [52]. The dependent evidence should be discounted in advance, and the discounting coefficients (or weight) are related to the degree of dependence. Guralnik et al. [53] presented a formal definition of algorithm dependency based on three criteria, i.e., method, sensors and features, and divided evidence into highly dependent, weakly dependent and independent evidence. Yager [54] proposed an interesting approach that makes use of a weighted aggregation of the belief structures where the weights are related to the degree of dependence. It is more practical to be used in real applications; however, how to define the degree of dependence is not addressed. To address this problem, Su et al. [43] presented a strategy of handling dependent evidence at a systematic level, which is able to capture both inner dependence (or interior relationship) and outer dependence (or exterior relationship). For inner dependence, they suggested using the analytic network process (ANP) to derive the degree of dependence. For outer dependence, they proposed a model based on the intersection of influencing factors identified during the information propagating and evaluating process. However, the method is subjective to some extent. For variables

with a certain amount of historical data and samples, statistical methods can be used to measure the dependence among information sources. Su et al. [53] suggests using the Pearson correlation coefficient to represent the correlation between evidence. However, the Pearson correlation coefficient presents only a linear correlation between two variables, which is not always the case in real applications. In this paper, we proposed a method to measure the dependence between evidence based on rank correlation coefficient that could remove the limitation of Pearson correlation coefficient.

This paper is organized as follows. In Section 2, the preliminaries on D–S evidence theory, the definition of the discounted BBA and Spearman’s rank correlation coefficient are briefly introduced. In Section 3, the model based on Spearman’s rank correlation coefficient is proposed. In Section 4, an experiment is illustrated to show the rationality of this new method. Finally, the conclusions are given in Section 5.

2. Preliminaries

Some preliminaries are introduced in this section, including Dempster–Shafer evidence theory, the discounted evidence, Pignistic Probability Transformation and Spearman’s rank correlation coefficient.

2.1. Dempster–Shafer Evidence Theory

Definition 1. Let $\Theta = \{A_1, A_2, \dots, A_N\}$ be a finite nonempty set of N elements that are mutual and exhaustive, and we define Θ as Frame of Discernment [43]. Let $P(\Theta)$ be the power set composed of 2^N elements of Θ . The Basic Belief Assignment function (BBA) is defined as a mapping from the power set $P(\Theta)$ to a number between 0 and 1, $m : P(\Theta) \rightarrow [0, 1]$, which satisfies the following conditions [43]:

$$m(\emptyset) = 0, \sum_{A \subseteq \Theta} m(A) = 1, \quad (1)$$

where $m(A)$ denotes the Basic Belief Assignment of proposition A .

Definition 2. (Dempster’s Rule) Let m_1, m_2, \dots, m_N be N independent BBAs in the frame of Discernment of Θ . The result of their combination is denoted as $m = m_1 \oplus m_2 \oplus \dots \oplus m_N$, and calculated as follows [16]:

$$\begin{cases} m(\emptyset) = 0, \\ m(A) = K^{-1} \sum_{\cap A_j = A, j=1}^N \prod_{i=1}^N m_i(A_j), \end{cases} \quad (2)$$

where K is normalizing factor, calculated as:

$$K = 1 - \sum_{\cap A_j = \emptyset, j=1}^N \prod_{i=1}^N m_i(A_j). \quad (3)$$

Definition 3. Let m be the BBA on Θ and α be the discount coefficient, $\alpha \in [0, 1]$ the discounted BBA ${}^\alpha m$ defined as:

$${}^\alpha m = \alpha \otimes m : \begin{cases} {}^\alpha m(A) = \alpha m(A), \forall A \subset \Theta, A \neq \Theta, \\ {}^\alpha m(\Theta) = 1 - \alpha + \alpha m(\Theta). \end{cases} \quad (4)$$

Definition 4. (Pignistic Probability Transformation, PPT) To make a decision after BBA fusion results in acquisition, there are two methods: the first method is decision according to the BBA fusion results, the second method is translating the BBA fusion results to the probability and making decision. In the first method, the information loss may be large, and the second method helps to draw a more accurate result. Based on such

consideration, Smets proposed the Pignistic Probability Transformation method [55]. Supposing m is BBA in Θ , let BetP be the Pignistic Probability distribution. The Pignistic Probability Transformation is defined as

$$\text{BetP}_m(\omega) = \sum_{A \subseteq \Theta, \omega \in A} \frac{1}{|A|} \frac{m(A)}{1 - m(\emptyset)}, m(\emptyset) \neq 1, \quad (5)$$

where $|A|$ is the cardinality of A , and \emptyset is denoted as the empty set.

2.2. Spearman's Rank Correlation Coefficient

There are all kinds of parameters to evaluate the dependent degree. The article [56] suggests using the Pearson correlation coefficient to represent the correlation between evidence. However, before using the Pearson correlation coefficient, it is necessary to assume that experiment data derived from normal distribution and was equidistant at least within the logical range. The rank correlation coefficient is a parameter-free measure for correlations that may be used to measure the level of agreement between two stochastic variables without making assumptions regarding the parametric structure of the probability distribution of the variables. The rank correlation coefficient, a parameter independent of the distribution, was proposed by Spearman in 1904 and used to measure the correlation between the two variables [57,58].

The basic idea of the Spearman rank correlation coefficient is to use the rank of the variable instead of the specific data for statistical inference [59]. Suppose that the variables x and y have n samples (measured values) denoted as x_i, y_i , where $i = 1, 2, \dots, n$. Sorting the sample data from large to small (or from small to large), let x'_i, y'_i be the position of original data x_i, y_i after arrangement. The Spearman rank correlation coefficient is defined as

$$r_s = \frac{\sum_{i=1}^n (x'_i - \bar{x}') (y'_i - \bar{y}')}{\sqrt{\sum_{i=1}^n (x'_i - \bar{x}')^2 \sum_{i=1}^n (y'_i - \bar{y}')^2}} = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}, \quad (6)$$

where $-1 \leq r_s \leq 1$, the $|r_s|$ is growing with x, y closer and closer to the strict monotonic function. $r_s = 1$ represents x, y becoming a strictly monotone increasing function and $r_s = -1$ represents x, y becoming a strictly monotone decreasing function. If $r_s = 0$, x, y have no relevance to the distinct monotonic function.

3. Proposed Method

3.1. The Framework of the Proposed Method

In this section, the method of handling dependent evidence is given in detail in order to fuse dependent sensor data properly. A flowchart of the proposed method is given in Figure 1. From Section 1, we can know that there are two basic directions to handle dependent evidence. One is to modify the Dempster's combination rule, finding a new evidence fusion method without considering dependence. The other is to reduce the repetitive computation of the dependent part of the information sources as far as possible. Here, we adopt the latter method to establish our evidence fusion model. First, collect sensor data as the raw data to generate BBA. Then, analyze the association between every two sensor information sources and calculate the correlation discount coefficient based on the analysis of sensor data. In addition, fuse the discounted BBA according to Dempster's combination rule. Finally, a decision conclusion is making by using the method of Pignistic Probability Transformation proposed in [55].

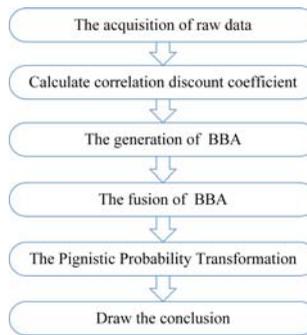


Figure 1. Flowchart of the proposed method.

3.2. The Generation of the Correlation Discount Coefficient

Thinking of the need to experience the construction and fusion of BBA from the sensor information source and the final decision making, the analysis of the relevance of the evidence should begin with the initial information source, as the real application involves multi-sensor sources of information, and the relationship between the sensor information source is complex. In order to find a simple and effective representation, this paper adopts the following method:

- Step1:** Calculate the Spearman rank correlation coefficient between every two sensor information sources according to Equation (6).
- Step2:** The correlation between the two sensor information sources can be divided into positive correlation and negative correlation. In other words, rank correlation coefficients are positive or negative, and negative correlation evidence can be regarded as conflict evidence. Calculate the dependence degree $d_{sx,sy}$ between two information sources:

$$d_{sx,sy} = |r_{sx,sy}|. \quad (7)$$

Here, $d_{sx,sy}$ is defined as the absolute value of $r_{sx,sy}$ because the positive correlation or negative correlation does not substantially affect the fusion result for the information fusion system based on the D-S evidence theory. To illustrate this problem, there is an example:

Suppose that there are two sensors A and B to qualitatively measure the water level. The recognition framework is defined as $\Theta = \{\text{high, middle, low}\}$. As shown in Figure 2, sensor A measures the distance between the top of the well and water surface, denoted as L_A , while sensor B measures the distance between the bottom of the well and water surface, denoted as L_B .

Assume that the depth of this well is 1. The interval $[0, 0.4]$, $[0.3, 0.7]$ and $[0.6, 1]$, respectively, represent the statuses of low water level, middle water level and high water level. By analysing the geometric relationship, L_A and L_B satisfy the follow equation:

$$L_A = -L_B + 1. \quad (8)$$

That is, L_A and L_B have a strictly negative correlation relationship, $r_{A,B} = -1$. However, L_A and L_B may have different values for the same water level, and they established the same BBA for the water level. For example, when $L_A = 0.9$, $L_B = 0.1$, both of the sensors establish the same BBA:

$$m(\{\text{low}\}) = 1. \quad (9)$$

In addition, when $L_A = 0.35, L_B = 0.65$, both of the sensors establish the same BBA:

$$m(\{middle\}, \{high\}) = 1. \tag{10}$$

In other words, the positive correlation or negative correlation does not substantially affect the fusion result for the information fusion system based on the D-S evidence theory.

Supposing that there are M sensor sources, we can then establish dependency matrix D:

$$D = \begin{bmatrix} d_{S1,S1} & d_{S1,S2} & \cdots & d_{S1,SM} \\ d_{S2,S1} & d_{S2,S2} & \cdots & d_{S2,SM} \\ \vdots & \vdots & \ddots & \vdots \\ d_{SM,S1} & d_{SM,S2} & \cdots & d_{SM,SM} \end{bmatrix}. \tag{11}$$

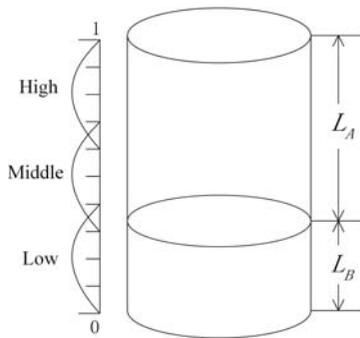


Figure 2. Measurement of the water level.

Step3: Calculate the total dependence degree of each sensor information source. The total dependence degree of S_i is defined as

$$T_{Si} = \sum_{k=1}^M d_{Si,Sk}. \tag{12}$$

Step4: Considering that the evidence with strong relevance should be given a smaller discount factor, the correlation discount coefficient is defined as

$$\alpha_{Si} = \frac{1}{T_{Si}}. \tag{13}$$

3.3. Reliability Assessment

In a multi-sensor information fusion system, the global system performance is closely related to each sensor’s reliability. The reliability of the information source refers to the correct rate obtained by direct decision by the information source. The higher reliability evidence helps to draw a more accurate result in the decision-making process. In addition, the higher reliability evidence should be given larger weight in evidence fusion. For the target recognition, it is assumed that there are N groups of training data to establish BBA by S_i and M of them have correct classification results. The reliability of the information source S_i is defined as

$$\beta_{Si} = \frac{M}{N}. \tag{14}$$

3.4. The Fusion of Dependent Evidence

On the basis of the above analyses, the total discounting coefficient of evidence from information source S_i , represented by ω_{S_i} , can be defined as

$$\omega_{S_i} = \alpha_{S_i} \beta_{S_i}. \quad (15)$$

Suppose that BBA $m_{S_1}, m_{S_2}, \dots, m_{S_M}$ is established by S_1, S_2, \dots, S_M , whose total discounting coefficient is $\omega_{S_1}, \omega_{S_2}, \dots, \omega_{S_M} \in [0, 1]$. The formula of dependent evidence fusion is as follows:

$$m = \omega_{S_1} m_{S_1} \oplus \omega_{S_2} m_{S_2} \oplus \dots \oplus \omega_{S_M} m_{S_M}, \quad (16)$$

where $\omega_{S_i} m_{S_i}$ represents discount calculation with discounting coefficient ω_{S_i} . (see Equation (4)).

4. Experiment and Discussion

In this paper, Iris Dataset from the machine learning database [60] is used as the data source of the experiment. The dataset contains three types of irises, such as Setosa, Versicolour and Virginica, and each group of irises contains 50 sets of data samples. Each group has four different attributes: SL (Sepal Length), SW (Sepal Width), PL (Petal Length) and PW (Petal Width). These four different attributes can be used as four kinds of information sources to construct BBA.

4.1. Possible Application of the Proposed Evidence Fusion Model

The dependent evidence fusion method in this paper could effectively deal with uncertain information fusion issues existing in the real world. One of the applications of this method could be target recognition. For example, to recognize the enemy aircraft, which could be in the form of a bomber, the Air Early Warning plane or fighter plane, we could first acquire information such as airborne radar signal, infrared signal or the electronic support measure (ESM) information. Different types of enemy aircraft usually have different features in the information. Thus, the type of aircraft could be recognized based on the acquired information. The recognition rate can be improved by combining information from different sensors. The dependence among the sensors is also considered in the proposed method. Similar to the iris example, the airborne radar signal, infrared signal or the ESM information are associated with the iris attributes SL, SW, PL or PW, and the types of enemy aircraft are associated with the iris types Setosa, Versicolour or Virginica.

Another possible application could be fault diagnosis. For example, in a power transformer fault diagnosis system, we want to recognize different fault types including transformer winding, transformer core, arc discharge or transformer insulation aging, and so on. In addition, the corresponding fault symptoms including transformer core earth current, insulation resistance or other symptoms could be acquired to establish different BBAs. Different fault symptoms may be dependent and the fusion method proposed in this article can handle this problem. In other words, the dependent information fusion model based on rank correlation coefficient could be investigated for applications in many fields.

4.2. Experimental Method

The main procedure of the proposed method to recognize iris class is shown in Figure 3.

In the experiment, some samples were randomly selected as the training set, and the remaining samples were used as test set. The steps are as follows:

- Step1:** Calculate the correlation discount coefficient α_{S_i} of four attributes. Refer to Section 3.2.
- Step2:** Build BBA. Four BBAs were established for SL, SW, PL and PW. The BBA was established according to the article [61].
- Step3:** Calculate reliability coefficient β_{S_i} , and refer to Section 3.3.

Step4: Calculate the total discounting coefficient of four attributes according to Equation (15).

Step5: Model testing. Using the test data, we assume for four cases that the evidence is independent from each other, only considering the dependence between the evidence, only considering the reliability of the evidence, considering both dependence and reliability, and then calculating the recognition accuracy. Then, we increase the proportion of training set data, and repeat the above experiment. In the four cases, the fusion rule as follows:

Group 1: Assume that the evidence is independent from each other

$$m = m_{S1} \oplus m_{S2} \oplus \dots \oplus m_{SM}. \tag{17}$$

Group 2: Only considering the dependence between the evidence

$$m = \alpha_{S1} m_{S1} \oplus \alpha_{S2} m_{S2} \oplus \dots \oplus \alpha_{SM} m_{SM}. \tag{18}$$

Group 3: Only considering the reliability of the evidence

$$m = \beta_{S1} m_{S1} \oplus \beta_{S2} m_{S2} \oplus \dots \oplus \beta_{SM} m_{SM}. \tag{19}$$

Group 4: Considering both dependence and reliability

$$m = \omega_{S1} m_{S1} \oplus \omega_{S2} m_{S2} \oplus \dots \oplus \omega_{SM} m_{SM}. \tag{20}$$

The method of decision-making after fusion is based on the Pignistic Probability Transformation (PPT) proposed in [55].

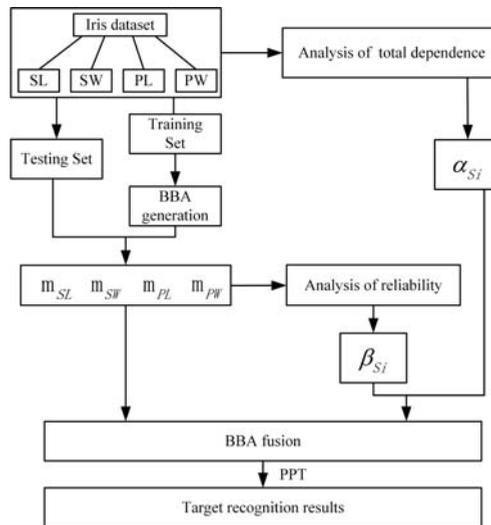


Figure 3. The main procedure of the proposed method to recognize iris class.

4.3. Experimental Procedure

In this section, the correlation of iris data sets is performed. The main procedures are as follows:

Step1: The first step is to analyse the dependence among SL, SW, PL, and PW attributes. Rank correlation coefficients among attributes of iris data sets are shown in Table 1.

Then, we can establish dependency matrix D:

$$D = \begin{bmatrix} 1.0000 & 0.1595 & 0.8814 & 0.8344 \\ 0.1595 & 1.0000 & 0.3034 & 0.2775 \\ 0.8814 & 0.3034 & 1.0000 & 0.9360 \\ 0.8344 & 0.2775 & 0.9360 & 1.0000 \end{bmatrix}.$$

Calculate the total dependence degree of each attribute as Equation (12). Results are as follows:

$$T_{SL} = 2.8753, T_{SW} = 1.7404, T_{PL} = 3.1208, T_{PW} = 3.0479.$$

Considering that the evidence with strong relevance should be given a smaller discount factor, the correlation discount coefficient is calculated as Equation (13). The results are as follows:

$$\alpha_{SL} = 0.3478, \alpha_{SW} = 0.5746, \alpha_{PL} = 0.3204, \alpha_{PW} = 0.3281.$$

Table 1. Correlation coefficients among attributes.

Attribute	SL	SW	PL	PW
SL	1.0000	-0.1595	0.8814	0.8344
SW	-0.1595	1.0000	-0.3034	-0.2775
PL	0.8814	-0.3034	1.0000	0.9360
PW	0.8344	-0.2775	0.9360	1.0000

Step2: Build BBA. Four BBAs were established for SL, SW, PL and PW. The BBA was established according to the article [61].

Step3: Calculate reliability coefficients of the four attributes as Equation (14). The results are as follows:

$$\beta_{SL} = 0.7267, \beta_{SW} = 0.5467, \beta_{PL} = 0.9533, \beta_{PW} = 0.9600.$$

Step4: The total discounting coefficient of the four attributes can be calculated as Equation (15). The results are as follows:

$$\omega_{SL} = 0.2527, \omega_{SW} = 0.3141, \omega_{PL} = 0.3054, \omega_{PW} = 0.3150.$$

After normalization, the total discounting coefficient of the four attributes is

$$\omega_{SL} = 0.8022, \omega_{SW} = 0.9971, \omega_{PL} = 0.9695, \omega_{PW} = 1.$$

Step5: Model testing. Based on the above calculation, we begin to test our evidence fusion model. The detailed steps are shown in Section 4.2.

Step6: Then, we calculate the confidence interval of this experiment. For each piece of evidence, we can acquire a BBA, and suppose its recognition result is A. Then, the Belief function (Bel(A)), defined as a sum of the mass probabilities of all the proper subsets of A, is calculated as follows:

$$Bel(A) = \sum_{B \subseteq A} m(B). \tag{21}$$

In addition, the Plausibility function (Pl(A)), defined as maximum belief of A, is calculated as follows:

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B). \tag{22}$$

Then, the Belief Interval is defined as $[Bel(A), PI(A)]$. Here, calculate the average Bel and PI of each proportion of testing data.

4.4. Results and Analysis

In four cases, the average classification recognition accuracy test results are shown in Figure 4.

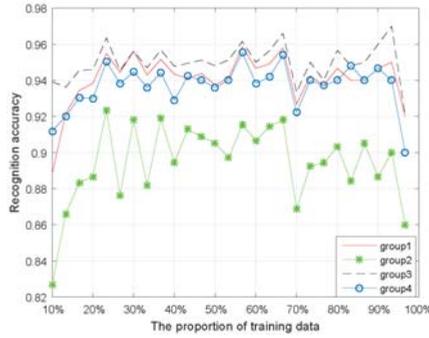
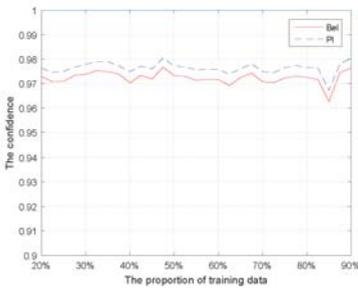
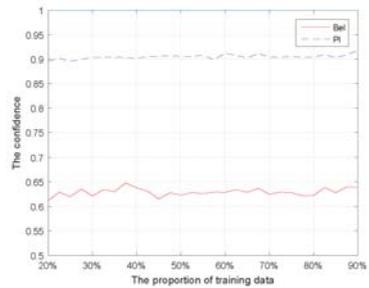


Figure 4. Average classification recognition accuracy in four cases.

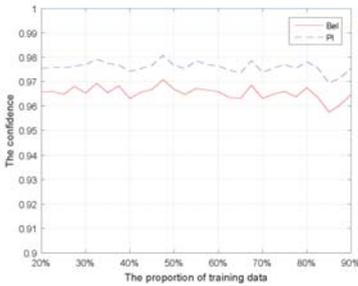
The confidence interval of this experiment as shown in Figure 5.



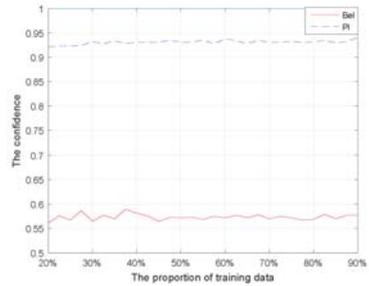
(a) the confidence interval of group 1.



(b) the confidence interval of group 2.



(c) the confidence interval of group 3.



(d) the confidence interval of group 4.

Figure 5. The confidence intervals of the experiments.

Comparing Group 1 with Group 3 (or Group 2 with Group 4), it is obvious that the recognition accuracy is significantly improved when considering reliability. However, comparing Group 1 with

Group 2 (or Group 3 with Group 4), the recognition accuracy is decreased. In addition, further research shows the reliability coefficients of the four attributes and the correlation coefficients in Table 2.

Table 2. The reliability coefficient and correlation coefficient of four attributes.

Coefficient	SL	SW	PL	PW
reliability coefficient	0.7267	0.5467	0.9533	0.9600
correlation coefficient	0.3478	0.5746	0.3204	0.3281

From Table 2, the reliability coefficient of PW or PL is large, indicating higher reliability. However, the correlation coefficient of PW or PL is small, indicating higher dependence of these two attributes in the whole system. The reliability coefficient of SW is small (lower reliability), while its correlation coefficient is the largest (the most independent attribute in the whole system). In this experiment, a higher recognition accuracy was obtained when using PW or PL attributes. However, there is strong dependence between PW and PL, and the higher recognition accuracy of PW or PL attributes is actually unreasonable for repeating fusion of two similar pieces of evidence. The recognition accuracy will be decreased when considering the effect of dependent evidence. On the other hand, the recognition accuracy will be below actual accuracy if two dependent pieces of evidence that both have low recognition accuracy are fused. In this case, the total recognition accuracy will increase when considering dependent evidence. In the next section, a case study is used to show the effectiveness of the proposed method.

4.5. Further Study

A case study is used to show the effectiveness of the proposed method. Assume that the framework is $\Theta = \{A, B, C\}$, and four independent pieces of evidence are R1, R2, R3 and R4. In this case, the correct recognition result is A. As are shown in the following table, and the pieces of evidence R1, R2, and R3 could draw the correct result; however, evidence R4 draws the wrong result (Table 3).

Table 3. Four pieces of evidence and their recognition results.

Item	BBA	PPT	Recognition Result
R1	$m(A) = 0.5, m(B) = 0.2, m(AC) = 0.3$	$p(A) = 0.65, p(B) = 0.2, p(C) = 0.15$	A
R2	$m(A) = 0.55, m(B) = 0.15, m(ABC) = 0.3$	$p(A) = 0.65, p(B) = 0.25, p(C) = 0.1$	A
R3	$m(A) = 0.61, m(AB) = 0.35, m(AC) = 0.04$	$p(A) = 0.805, p(B) = 0.175, p(C) = 0.02$	A
R4	$m(AC) = 0.15, m(B) = 0.55, m(BC) = 0.3$	$p(A) = 0.075, p(B) = 0.7, p(C) = 0.225$	B

Case1: Combining these four independent pieces of evidence according to the traditional Dempster's rule:

$$m = m_{R1} \oplus m_{R2} \oplus m_{R3} \oplus m_{R4}$$

after PPT, the probability is

$$p(A) = 0.7834, p(B) = 0.2062, p(C) = 0.0104.$$

The recognition result is A, which is correct by combining four independent pieces of evidence.

Case2: Let us consider another condition: we have the fifth piece of evidence R5, which is the same as R4, that is, $R5 : m(AC) = 0.15, m(B) = 0.55, m(BC) = 0.3$. Apparently, R4 is totally dependent on R5, and there may be a mistake if we combine these five pieces of evidence regardless of their association; for example:

$$m = m_{R1} \oplus m_{R2} \oplus m_{R3} \oplus m_{R4} \oplus m_{R5}.$$

after PPT, and the probability is

$$p(A) = 0.3942, p(B) = 0.5880, p(C) = 0.0178.$$

The recognition result is B; therefore, using traditional Dempster's rule without considering dependent can draw a wrong result. The reason is that one of the pieces of evidence is totally dependent on another piece of evidence, which means that one of the pieces of evidence has been counted twice.

Case3: By considering the dependent evidence based on the method proposed in Section 3, the correlation discount coefficient could be given as follows:

$$\alpha_{R1} = 1, \alpha_{R2} = 1, \alpha_{R3} = 1, \alpha_{R4} = 0.5, \alpha_{R5} = 0.5.$$

Then, the fusion rule is

$$m = \alpha_{R1} m_{R1} \oplus \alpha_{R2} m_{R2} \oplus \alpha_{R3} m_{R3} \oplus \alpha_{R4} m_{R4} \oplus \alpha_{R5} m_{R5}.$$

The final result is

$$p(A) = 0.8881, p(B) = 0.1068, p(C) = 0.0051.$$

The recognition result is A. Thus, the proposed dependent evidence fusion model could improve the decision-making result especially in the fusion of low recognition evidence.

5. Conclusions

With the rapid development of artificial intelligence, the acquisition of information has increasing importance. In industrial applications, the multi-sensor information fusion system based on Dempster-Shafer evidence theory plays a more and more important role in information collection and decision-making. However, the classical evidence theory assumes that the evidence is independent from each other, which is often difficult to establish in practice. To address this issue, this paper analyzes the present research about dependent evidence fusion. Comparing with all kinds of correlation measurement methods, we select the Spearman's rank correlation coefficient as the metric to measure dependence existing in evidence. Rank correlation coefficient is a parameter-free measure for correlations, which may be used to measure the level of agreement between two stochastic variables without making assumptions regarding the parametric structure of the probability distribution of the variables. Then, a dependent evidence fusion model based on rank correlation coefficient is established. Finally, an experiment is developed to verify the effectiveness of this model based on iris data sets. Experiment results suggest that considering reliability will improve accuracy of decision-making and considering dependent evidence helps draw more reasonable and robust conclusions. In other words, it is necessary to consider the influence of dependent evidence in information fusion to gain a more credible result.

Acknowledgments: This work was partially supported by the National Natural Science Foundation of China (Grant No. 61503237, Grant No. 61573290), the Shanghai Natural Science Foundation (Grant No. 15ZR1418300), the Shanghai Science and Technology Commission Key Program (Grant No. 15160500800), the Shanghai Key Laboratory of Power Station Automation Technology (No. 13DZ2273800), and the Shanghai Education Commission Excellent Youth Project (Grant No. ZZsd115144).

Author Contributions: F.S. and X.S. designed and performed the research; F.S. wrote the paper; H.Q. and N.Y. performed the computation; X.S. and F.S. analyzed the data; and W.H. contributed analysis tools. All authors discussed the results and commented on the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tian, T.; Sun, S.; Li, N. Multi-sensor information fusion estimators for stochastic uncertain systems with correlated noises. *Inf. Fusion* **2016**, *27*, 126–137.
2. Tang, Y.; Zhou, D.; Xu, S.; He, Z. A weighted belief entropy-based uncertainty measure for multi-sensor data fusion. *Sensors* **2017**, *17*, 928.
3. Villarrubia, G.; Paz, J.F.D.; Iglesia, D.H.D.L.; Bajo, J. Combining multi-agent systems and wireless sensor networks for monitoring crop irrigation. *Sensors* **2017**, *17*, 1775.
4. Poria, S.; Cambria, E.; Bajpai, R.; Hussain, A. A review of affective computing: From unimodal analysis to multimodal fusion. *Inf. Fusion* **2017**, *37*, 98–125.
5. Poria, S.; Cambria, E.; Howard, N.; Huang, G.B.; Hussain, A. Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing* **2016**, *174*, 50–59.
6. Cambria, E.; White, B. Jumping NLP Curves: A Review of Natural Language Processing Research. *IEEE Comput. Intell. Mag.* **2014**, *9*, 48–57.
7. Poria, S.; Cambria, E.; Gelbukh, A. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowl. Based Syst.* **2016**, *108*, 42–49.
8. Jin, X.B.; Zheng, H.-J.; Du, J.-J.; Wang, Y.-M. S-Retinex brightness adjustment for night-vision image. **2011**, *15*, 2571–2576.
9. Hao, G.; Sun, S.L.; Li, Y. Nonlinear weighted measurement fusion unscented kalman filter with asymptotic optimality. *Inf. Sci.* **2015**, *299*, 85–98.
10. Alexandridis, A. Evolving RBF neural networks for adaptive soft-sensor design. *Int. J. Neural Syst.* **2013**, *23*, 1350029.
11. Chen, S.; Deng, Y.; Wu, J. Fuzzy sensor fusion based on evidence theory and its application. *Appl. Artif. Intell.* **2013**, *27*, 235–248.
12. Frikha, A.; Moalla, H. Analytic hierarchy process for multi-sensor data fusion based on belief function theory. *Eur. J. Oper. Res.* **2015**, *241*, 133–147.
13. Deng, X.; Jiang, W.; Zhang, J. Zero-sum matrix game with payoffs of Dempster-Shafer belief structures and its applications on sensors. *Sensors* **2017**, *17*, 922.
14. Jiang, W.; Hu, W.; Xie, C. A new engine fault diagnosis method based on multi-sensor data fusion. *Appl. Sci.* **2017**, *7*, 280.
15. Jiang, W.; Zhuang, M.; Xie, C. A reliability-based method to sensor data fusion. *Sensors* **2017**, *17*, 1575.
16. Dempster, A. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339.
17. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
18. Cattaneo, M.E.G.V. Belief functions combination without the assumption of independence of the information sources. *Knowl. Based Syst.* **2011**, *52*, 299–315.
19. Wang, Y.M.; Yang, J.B.; Xu, D.L.; Chin, K.S. On the combination and normalization of interval-valued belief structures. *Inf. Sci. Int. J.* **2007**, *177*, 1230–1247.
20. Yang, Y.; Han, D. A new distance-based total uncertainty measure in the theory of belief functions. *Knowl. Based Syst.* **2016**, *94*, 114–123.
21. Deng, Y. Generalized evidence theory. *Appl. Intell.* **2015**, *43*, 530–543.
22. Cheng, G.; Chen, X.H.; Shan, X.L.; Liu, H.G.; Zhou, C.F. A new method of gear fault diagnosis in strong noise based on multi-sensor information fusion. *J. Vib. Control.* **2016**, *22*, 1504–1515.
23. Zhang, J.C.; Zhou, S.P.; Li, Y.; Su, Y.S.; Zhang, P.G. Improved D–S evidence theory for pipeline damage identification. *J. Vibroeng.* **2015**, *17*, 3527–3537.
24. Liu, Z.G.; Pan, Q.; Dezert, J.; Martin, A. Adaptive imputation of missing values for incomplete pattern classification. *Pattern Recognit.* **2016**, *52*, 85–95.
25. Liu, Z.G.; Pan, Q.; Dezert, J.; Mercier, G. Credal c-means clustering method based on belief functions. *Knowl. Based Syst.* **2015**, *74*, 119–132.
26. Denoeux, T. A neural network classifier based on Dempster–Shafer theory. *IEEE Trans. Syst. Man Cybern. Syst.* **2000**, *30*, 131–150.
27. Kang, B.; Deng, Y.; Sadiq, R.; Mahadevan, S. Evidential cognitive maps. *Knowl. Based Syst.* **2012**, *35*, 77–86.

28. Wei, D.; Deng, X.; Zhang, X.; Deng, Y.; Mahadevan, S. Identifying influential nodes in weighted networks based on evidence theory. *Phys. A Stat. Mech. Its App.* **2013**, *392*, 2564–2575.
29. Yager, R.R. Set measure directed multi-Source information fusion. *IEEE Tran. Fuzzy Syst.* **2011**, *19*, 1031–1039.
30. Yager, R.R. A knowledge-based approach to adversarial decision making. *Int. J. Intell. Syst.* **2008**, *23*, 1–21.
31. Yang, J.B.; Xu, D.L. Nonlinear information aggregation via evidential reasoning in multiattribute decision analysis under uncertainty. *IEEE Tran. Syst. Man Cybern. Syst.* **2002**, *32*, 376–393.
32. Yang, J.; Wang, Y.; Xu, D.; Chin, K. The evidential reasoning approach for MADA under both probabilistic and fuzzy uncertainties. *Eur. J. Oper. Res.* **2006**, *171*, 309–343.
33. Ma, W.; Xiong, W.; Luo, X. A model for decision making with missing, imprecise, and uncertain evaluations of multiple criteria. *Int. J. Intell. Syst.* **2013**, *28*, 152–184.
34. Yao, S.; Huang, W.Q. Induced ordered weighted evidential reasoning approach for multiple attribute decision analysis with uncertainty. *Int. J. Intell. Syst.* **2014**, *29*, 906–925.
35. Huang, S.; Su, X.; Hu, Y.; Mahadevan, S.; Deng, Y. A new decision-making method by incomplete preferences based on evidence distance. *Knowl. Based Syst.* **2014**, *56*, 264–272.
36. Wang, X.; Yang, F.; Li, D.; Zhang, F. Group decision-making method for dynamic cloud model based on cumulative prospect theory. *Int. J. Grid Distrib. Comput.* **2016**, *9*, 283–290.
37. Yang, J.; Huang, H.Z.; He, L.P.; Zhu, S.P.; Wen, D. Risk evaluation in failure mode and effects analysis of aircraft turbine rotor blades using Dempster–Shafer evidence theory under uncertainty. *Eng. Fall. Anal.* **2011**, *18*, 2084–2092.
38. Wang, X.; Yang, F.; Wei, H.; Ji, L. A risk assessment model of uncertainty system based on set-valued mapping. *J. Intell. Fuzzy Syst.* **2016**, *31*, 3155–3162.
39. Ji, L.; Yang, F.; Wang, X. Transformation of possibility distribution into mass function and method of ordered reliability decision. *J. Comput. Theor. Nanosci.* **2016**, *13*, 4454–4460.
40. Su, X.; Mahadevan, S.; Xu, P.; Deng, Y. Dependence assessment in human reliability analysis using evidence theory and AHP. *Risk Anal.* **2015**, *35*, 1296–1316.
41. Shafer, G. The problem of dependent evidence. *Int. J. Approx. Reason.* **2016**, *79*, 41–44.
42. Denooux, T. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artif. Intell.* **2008**, *172*, 234–264.
43. Su, X.; Mahadevan, S.; Xu, P.; Deng, Y. Handling of dependence in Dempster–Shafer theory. *Int. J. Intell. Syst.* **2015**, *30*, 441–467.
44. Su, X.; Han, W.; Xu, P.; Deng, Y. Review of combining dependent evidence. *Syst. Eng. Electron.* **2016**, *38*, 1345–1351.
45. Su, X.; Mahadevan, S.; Han, W.; Deng, Y. Combining dependent bodies of evidence. *Appl. Intell.* **2016**, *44*, 634–644.
46. Cattaneo, Marco EGV. Combining belief functions issued from dependent sources. In Proceedings of the Third International Symposium on Imprecise Probabilities and Their Application (isipta'03), Lugano, Switzerland, 14–17 July 2003; pp. 133–147.
47. Destercke, S.; Dubois, D.; Chojnacki, E. Cautious conjunctive merging of belief functions. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 9th European Conference, ECSQARU 2007, Hammamet, Tunisia, 31 October 31–2 November 2007*; Mellouli, K., Ed.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 332–343.
48. Destercke, S.; Dubois, D. Idempotent conjunctive combination of belief functions: Extending the minimum rule of possibility theory. *Inf. Sci.* **2011**, *181*, 3925–3945.
49. Choenni, S.; Blok, H.E.; Leertouwer, E. Handling uncertainty and ignorance in databases: A rule to combine dependent data. In *Database Systems for Advanced Applications: 11th International Conference, DASFAA 2006, Singapore, 12–15 April 2006*; Li Lee, M., Tan, K.L., Wuwongse, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 310–324.
50. Chebbah, M.; Martin, A.; Ben Yaghlane, B. Combining partially independent belief functions. *Decis. Support Syst.* **2015**, *73*, 37–46.
51. Smets, P. The concept of distinct evidence. In Proceedings of the 4th Conference on Information Processing and management of Uncertainty in Knowledge-based Systems (IPMU), Palma de Mayorca, Spain, 6–10 July 1992; pp. 789–794.

52. Jiang, H.N.; Xu, X.B.; Wen, C.L. The combination method for dependent evidence and its application for simultaneous faults diagnosis. In Proceedings of the 2009 International Conference on Wavelet Analysis and Pattern Recognition, Baoding, China, 12–15 July 2009; pp. 496–501.
53. Guralnik, V.; Mylaraswamy, D.; Voges, H. On handling dependent evidence and multiple faults in knowledge fusion for engine health management. In Proceedings of the 2006 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2006; pp. 9–10.
54. Yager, R.R. On the fusion of non-independent belief structures. *Int. J. Gen. Syst.* **2009**, *38*, 505–531.
55. Smets, P.; Kennes, R. The transferable belief model. *Artif. Intell.* **1994**, *66*, 191–234.
56. Su, X.; Xu, P.; Mahadevan, S.; Deng, Y. On consideration of dependence and reliability of evidence in Dempster–Shafer theory. *J. Inf. Comput. Sci.* **2014**, *11*, 4901–4910.
57. Bishara, A.J.; Hittner, J.B. Testing the significance of a correlation with nonnormal data: Comparison of Pearson, Spearman, transformation, and resampling approaches. *Psychol. Methods* **2012**, *17*, 399–417.
58. Puth, M.T.; Neuhauser, M.; Ruxton, G.D. Effective use of Spearman’s and Kendall’s correlation coefficients for association between two measured traits. *Anim. Behav.* **2015**, *102*, 77–84.
59. Zhang, W.Y.; Wei, Z.W.; Wang, B.H.; Han, X.P. Measuring mixing patterns in complex networks by Spearman rank correlation coefficient. *Physica A* **2016**, *451*, 440–450.
60. UCI Machine Learning Repository. Available online: <http://archive.ics.uci.edu/ml/datasets/Iris> (accessed on 10 August 2017).
61. Xu, P.; Deng, Y.; Su, X.; Mahadevan, S. A new method to determine basic probability assignment from training data. *Knowl. Based Syst.* **2013**, *46*, 69–80.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

A Novel Evidence Theory and Fuzzy Preference Approach-Based Multi-Sensor Data Fusion Technique for Fault Diagnosis

Fuyuan Xiao

School of Computer and Information Science, Southwest University, No. 2 Tiansheng Road, BeiBei District, Chongqing 400715, China; xiaofuyuan@swu.edu.cn

Received: 12 September 2017; Accepted: 27 October 2017; Published: 31 October 2017

Abstract: The multi-sensor data fusion technique plays a significant role in fault diagnosis and in a variety of such applications, and the Dempster–Shafer evidence theory is employed to improve the system performance; whereas, it may generate a counter-intuitive result when the pieces of evidence highly conflict with each other. To handle this problem, a novel multi-sensor data fusion approach on the basis of the distance of evidence, belief entropy and fuzzy preference relation analysis is proposed. A function of evidence distance is first leveraged to measure the conflict degree among the pieces of evidence; thus, the support degree can be obtained to represent the reliability of the evidence. Next, the uncertainty of each piece of evidence is measured by means of the belief entropy. Based on the quantitative uncertainty measured above, the fuzzy preference relations are applied to represent the relative credibility preference of the evidence. Afterwards, the support degree of each piece of evidence is adjusted by taking advantage of the relative credibility preference of the evidence that can be utilized to generate an appropriate weight with respect to each piece of evidence. Finally, the modified weights of the evidence are adopted to adjust the bodies of the evidence in the advance of utilizing Dempster’s combination rule. A numerical example and a practical application in fault diagnosis are used as illustrations to demonstrate that the proposal is reasonable and efficient in the management of conflict and fault diagnosis.

Keywords: sensor data fusion; evidential conflict; evidence distance; belief entropy; variance of entropy; fuzzy preference relations; Dempster–Shafer evidence theory; fault diagnosis

1. Introduction

The multi-sensor data fusion technique plays a significant role in fault diagnosis. Due to the complexity of the targets, the report collected from a single sensor is insufficient in decision making processes. Additionally, because of the impact of the environment, the data gathered from multiple sensors may be unreliable or even wrong so that it can cause erroneous results in fault diagnosis. Hence, multi-sensor data fusion technologies are required in various fields of practical applications [1–9], especially in the area of data fusion using vibration data [10–15]. However, in the practical applications, the data that are gathered from the multi-sensors are usually uncertain. An open issue is how to model and handle such kinds of uncertain information. To address this issue, a variety of theoretical methods has been exploited for multi-sensor data fusion, like the rough sets theory [16,17], fuzzy sets theory [18–22], evidence theory [23–25], Z numbers [26,27], and D numbers theory [28–30], evidential reasoning [31–34], and so on [35–38].

Dempster–Shafer evidence theory, which is an uncertainty reasoning tool, was firstly proposed by Dempster [23]; then, it was developed by Shafer [24]. Because Dempster–Shafer evidence theory is flexible and effective in modeling the uncertainty regardless of prior information, it is widely applied to various areas of information fusion, like pattern recognition [39–41],

decision making [42–47], supplier selection [48,49], optimization problems [50,51], risk analysis [52–54] and fault diagnosis [55–60]. Although Dempster–Shafer evidence theory has many advantages, it may generate counter-intuitive results, when fusing highly conflicting pieces of evidence [61,62]. To solve this problem, many methods have been proposed. They are divided into two types of methodologies [63–67]. The first type involves modifying Dempster’s combination rule, while the second type involves pre-processing the bodies of evidence. The main research works for the first type include the unnormalized combination rule presented by Smets [68], the disjunctive combination rule proposed by Dubois and Prade [69] and the combination rule presented by Yager [70]. Nevertheless, the modification of the combination rule often destructs the good properties, like the commutativity and associativity. Furthermore, if sensor failure results in the counter-intuitive results, such a modification is regarded to be unreasonable. Therefore, many researchers pre-process the bodies of evidence to resolve the problem of highly conflicting evidence, which falls into the second type. The main research works for the second type include the simple average approach of the bodies of evidence proposed by Murphy [71], the weighted average of the masses based on the evidence distance presented by Deng et al. [72] and the cosine theorem-based method proposed by Zhang et al. [73]. Deng et al.’s weighted average approach [72] overcomes the weakness of Murphy’s method [71] to some extent. Later on, Zhang et al. [73] made an improvement based on [72] and introduced the concept of vector space to handle the conflicting evidence. However, the effect of evidence’s uncertainty itself on the weight was overlooked.

In this paper, therefore, a novel multi-sensor data fusion method is proposed, which is a hybrid methodology in terms of the distance of evidence, belief entropy and fuzzy preference relation analysis. The proposal considers the support degree among the pieces of evidence, the uncertainty measure of the evidence and the effect of the relative credibility of the evidence on the weight, so that it can obtain more appropriately weighted average evidence before using Dempster’s combination rule. Specifically, the proposed method consists of the following procedures. First, in order to measure the support degree between the pieces of evidence, the function of evidence distance is leveraged, where the support degree represents the reliability of the evidence. After that, the relative credibility preference of the evidence is indicated by taking advantage of the fuzzy preference relation analysis on the foundation of the uncertainty of each piece of evidence measured by the belief entropy. Based on that, the support degrees of the evidence are adjusted, which can be utilized to generate the appropriate weights with regard to the evidence. Finally, the weighted average evidence can be obtained on the basis of the modified weights of the evidence before using Dempster’s combination rule. A numerical example and a practical application in fault diagnosis are used as illustrations to demonstrate that the proposed method outperforms the related methods with respect to the conflict management and fault diagnosis.

The remaining content of this paper is arranged below. Section 2 introduces the preliminaries of this paper briefly. In Section 3, a novel multi-sensor data fusion approach with regard to fault diagnosis is proposed. Section 4 gives a numerical example to illustrate the effectiveness of the proposal. Then, the proposed method is applied to a practical application in fault diagnosis in Section 5. Finally, Section 6 gives the conclusion.

2. Preliminaries

2.1. Dempster–Shafer Evidence Theory

Dempster–Shafer evidence theory [23,24] is extensively applied to handle uncertain information that belongs to the category of artificial intelligence. Because Dempster–Shafer evidence theory is flexible and effective in modeling the uncertainty regardless of prior information, it requires weaker conditions compared with the Bayesian theory of probability. When the probability is confirmed, Dempster–Shafer evidence theory degenerates to the probability theory and is considered as a generalization of Bayesian inference [74]. In addition, Dempster–Shafer evidence theory has the

advantage that it can directly express the “uncertainty” via allocating the probability into the set’s subsets, which consists of multi-objects, instead of a single object. Furthermore, it is capable of combining the bodies of evidence to derive new evidence. The basic concepts and definitions are described as below.

Definition 1 (Frame of discernment). Let Θ be a nonempty set of events that are mutually-exclusive and collectively-exhaustive, defined by:

$$\Theta = \{F_1, F_2, \dots, F_i, \dots, F_N\}, \tag{1}$$

in which the set Θ denotes a frame of discernment.

The power set of Θ is represented as 2^Θ , where:

$$2^\Theta = \{\emptyset, \{F_1\}, \{F_2\}, \dots, \{F_N\}, \{F_1, F_2\}, \dots, \{F_1, F_2, \dots, F_i\}, \dots, \Theta\}, \tag{2}$$

and \emptyset is an empty set.

When A is an element of the power set of Θ , i.e., $A \in 2^\Theta$, A is called a hypothesis or proposition.

Definition 2 (Mass function). In the frame of discernment Θ , a mass function m is represented as a mapping from 2^Θ to $[0, 1]$ that is defined as:

$$m : 2^\Theta \rightarrow [0, 1], \tag{3}$$

which meets the conditions below:

$$\begin{aligned} m(\emptyset) &= 0, \\ \sum_{A \in 2^\Theta} m(A) &= 1. \end{aligned} \tag{4}$$

The mass function m in the Dempster–Shafer evidence theory can also be called a basic probability assignment (BPA). When $m(A)$ is greater than zero, A as the element of 2^Θ is named as a focal element of the mass function, where the mass function $m(A)$ indicates how strongly the evidence supports the proposition or hypothesis A .

Definition 3 (Belief function). Let A be a proposition where $A \subseteq \Theta$; the belief function Bel of the proposition A is defined by:

$$\begin{aligned} Bel : 2^\Theta &\rightarrow [0, 1], \\ Bel(A) &= \sum_{B \subseteq A} m(B). \end{aligned} \tag{5}$$

The plausibility function Pl of the proposition A is defined by:

$$\begin{aligned} Pl : 2^\Theta &\rightarrow [0, 1], \\ Pl(A) &= 1 - Bel(\bar{A}) = \sum_{B \cap A \neq \emptyset} m(B), \end{aligned} \tag{6}$$

where \bar{A} is the complement of A , such that $\bar{A} = \Theta - A$.

Apparently, the plausibility function $Pl(A)$ is equal to or greater than the belief function $Bel(A)$, where the belief function Bel is the lower limit function of the proposition A , and the plausibility function Pl is the upper limit function of the proposition A .

Definition 4 (Dempster’s rule of combination). Let two basic probability assignments (BPAs) be m_1 and m_2 on the frame of discernment Θ where the BPAs m_1 and m_2 are independent; Dempster’s rule of combination, defined by $m = m_1 \oplus m_2$, which is called the orthogonal sum, is represented as below:

$$m(A) = \begin{cases} \frac{1}{1-K} \sum_{B \cap C = A} m_1(B)m_2(C), & A \neq \emptyset, \\ 0, & A = \emptyset, \end{cases} \tag{7}$$

with:

$$K = \sum_{B \cap C = \emptyset} m_1(B)m_2(C), \tag{8}$$

where B and C are also the elements of 2^Θ and K is a constant that presents the conflict between the BPAs m_1 and m_2 .

Notice that Dempster’s combination rule is only practicable for the BPAs m_1 and m_2 under the condition that $K < 1$.

2.2. Distance of Pieces of Evidence

Jousselme et al. [75] presented a distance function of the evidence to measure the distance among the basic probability assignments (BPAs), which is defined as below.

Definition 5 (Distance between two BPAs). Let two basic probability assignments (BPAs) m_1 and m_2 be on the same frame of discernment Θ , which contains N number of mutually-exclusive and collectively-exhaustive propositions. The distance between the BPAs m_1 and m_2 is denoted as:

$$d(m_1, m_2) = \sqrt{\frac{1}{2}(\vec{m}_1 - \vec{m}_2)^T \underline{D}(\vec{m}_1 - \vec{m}_2)}, \tag{9}$$

where $\vec{m}_i (i = 1, 2)$ is a 2^N -dimensional column vector and \underline{D} is a $(2^N \times 2^N)$ -dimensional matrix.

The elements of \underline{D} that measure the conflict of the focal elements in the BPAs m_1 and m_2 can be represented as:

$$\underline{D}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{10}$$

where $A \in 2^\Theta$ and $B \in 2^\Theta$. $|A \cap B|$ denotes the amount of objects in common between the elements A and B , while $|A \cup B|$ represents the subset’s cardinality of the union elements A and B .

It can be stated that $\underline{D}(A, B) \in [0, 1]$. Specifically, the value of $|A \cap B|$ is zero, when no common items exist between the elements A and B , which means that the element A highly conflicts with the element B so that the degree of similarity between the elements A and B is zero. Therefore, the smaller the $\underline{D}(A, B)$ is, the less similarity between the elements A and B there is; whereas, $\underline{D}(A, B) = 1$ indicates that the element A is identical to the element B .

2.3. Belief Entropy

A belief entropy, called the Deng entropy, was first proposed by Deng [43] and has been applied in various fields [76]. As the generalization of the Shannon entropy [77,78], the Deng entropy is an effective math tool for measuring the uncertain information, because the uncertain information can be expressed by BPAs, so that it can be used in the evidence theory. In such a situation that the uncertainty is expressed by the probability distribution, the uncertain degree measured by the Deng entropy will be identical to the uncertain degree measured by the Shannon entropy. The basic concepts and definitions are introduced below.

Let A be a proposition of the basic probability assignment (BPA) m on the frame of discernment Θ ; the Deng entropy $E_d(m)$ of the BPA m is defined as follows:

$$E_d(m) = - \sum_{A \subseteq \Theta} m(A) \log \frac{m(A)}{2^{|A|} - 1}, \tag{11}$$

where $|A|$ is the cardinality of the proposition A .

When the belief is only allocated to the single object, which means that $|A| = 1$, the Deng entropy degenerates to the Shannon entropy, namely,

$$E_d(m) = - \sum_{A \in \Theta} m(A) \log \frac{m(A)}{2^{|A|} - 1} = - \sum_{A \in \Theta} m(A) \log m(A). \tag{12}$$

The larger the cardinality of the proposition is, the larger the Deng entropy of evidence is, so that the evidence contains more information. When a piece of evidence has a big Deng entropy, it is supposed to be better supported by other evidence, which represents that this evidence plays an important part in the final combination.

2.4. Fuzzy Preference Relations

Fuzzy preference relations play a fundamental part in many decision-making processes, and they were first presented by Tanino [79] in 1984. It is a kind of method that can construct the decision matrices of pairwise comparisons by using the linguistic values that are provided by experts. The basic concepts are introduced below.

Definition 6 (Fuzzy preference relations [79–82]). *Let P be a fuzzy preference relation and $X = \{A_1, A_2, \dots, A_k\}$ be a set of alternatives, where $X \subseteq \Theta$, then the fuzzy preference relation is defined as follows:*

$$P = (p_{ij})_{k \times k} = \begin{bmatrix} 0.5 & \cdots & p_{1i} & \cdots & p_{1k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{i1} & \cdots & 0.5 & \cdots & p_{ik} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{k1} & \cdots & p_{ki} & \cdots & 0.5 \end{bmatrix}, \tag{13}$$

where $p_{ij} \in [0, 1]$ ($1 \leq i \neq j \leq k$) represents the preference value for the alternative A_i over A_j , which meets the conditions below:

$$p_{ij} + p_{ji} = 1 \text{ and } p_{ii} = 0.5. \tag{14}$$

It should be stated that $p_{ij} = 0.5$ represents the indifference between the alternatives A_i and A_j ; $p_{ij} = 1$ represents that A_i is absolutely preferred by A_j ; $p_{ij} > 0.5$ represents that A_i is preferred by A_j .

Whereas, the preference values may be inconsistent in the fuzzy preference relation, hence, Tanino [79] proposed the concept of the additive consistency for the fuzzy preference relation $P = (p_{ij})_{k \times k}$ as follows:

$$p_{ir} = p_{ij} + p_{jr} - 0.5, \tag{15}$$

where $p_{ii} = 0.5$ and $p_{ij} + p_{ji} = 1$ ($1 \leq i \neq j \neq r \leq k$).

After that, Lee [80] claimed that the complete fuzzy preference relation may not satisfy the consistency of the order in certain cases. Hence, the consistency of the order in fuzzy preference relations was presented by Lee [80] to solve this problem.

Definition 7 (The consistency matrix [80]). *Let $P^* = (p_{ij})_{k \times k}$ be a complete fuzzy preference relation, in which p_{ij} represents the preference values for the alternative A_i over A_j , $p_{ij} + p_{ji} = 1$ and $p_{ii} = 0.5$*

($1 \leq i \neq j \leq k$). The consistency matrix \bar{P} can be constructed on the basis of the complete fuzzy preference relation P^* , which is defined by:

$$\bar{P} = (\bar{p}_{ir})_{k \times k} = \left(\frac{1}{k} \sum_{j=1}^k (p_{ij} + p_{jr}) - 0.5 \right)_{k \times k}. \tag{16}$$

The consistency matrix $\bar{P} = (\bar{p}_{ir})_{k \times k}$ ($1 \leq i \neq r \leq k$) has the properties below:

- (1) $\bar{p}_{ir} + \bar{p}_{ri} = 1$;
- (2) $\bar{p}_{ii} = 0.5$;
- (3) $\bar{p}_{ir} = \bar{p}_{ij} + \bar{p}_{jr} - 0.5$;
- (4) $\bar{p}_{ir} \leq \bar{p}_{is}$ for all $i \in \{1, 2, \dots, k\}$, where $s \neq i$ and $s \neq r$.

Let $\bar{P} = (\bar{p}_{ir})_{k \times k}$ be a consistency matrix; the ranking value of alternative A_i , denoted as $RV(A_i)$, is defined by:

$$RV(A_i) = \frac{2}{k^2} \sum_{j=1}^k \bar{p}_{ij}, \tag{17}$$

where $1 \leq i \leq k$ and $\sum_{i=1}^k RV(A_i) = 1$.

3. The Proposed Method

In this paper, by considering not only the conflicts between pieces of evidence, but also the impact of the evidence’s uncertainty itself, a novel multi-sensor data fusion approach is presented and applied in fault diagnosis. The proposed method is a hybrid methodology that integrates the distance of evidence, belief entropy and fuzzy preference relation analysis, which consists of the following parts.

The function of evidence distance is first leveraged for measuring the conflict degree among the pieces of evidence, then the support degree resulting from the distance of the evidence is obtained to denote the evidence’s reliability. When the evidence is well supported by other pieces of evidence, it is supposed to have less conflict with other pieces of evidence, so that a big weight should be allocated to this piece of evidence. Instead, when the evidence is poorly supported by other pieces of evidence, it is regarded to highly conflict with other pieces of evidence so that a small weight should be allocated to this evidence. Next, the information volume of the evidence is calculated by making use of the belief entropy. Based on the calculated quantitative information volume, the fuzzy preference relations analysis is applied to indicate the relative credibility preference in terms of the pieces of evidence. Whereafter, the support degree of the evidence is adjusted by taking advantage of the relative credibility preference of the pieces of evidence. Thanks to introducing the fuzzy preference relations analysis based on the belief entropy, it can automatically construct the fuzzy preference relation matrix, rather than being determined by experts, which decreases the epistemic non-determinacy in the decision-making process. Finally, the adjusted weights of the pieces of evidence are applied to modify the body of the pieces of evidence before utilizing Dempster’s combination rule. The flowchart of the proposal is shown in Figure 1.

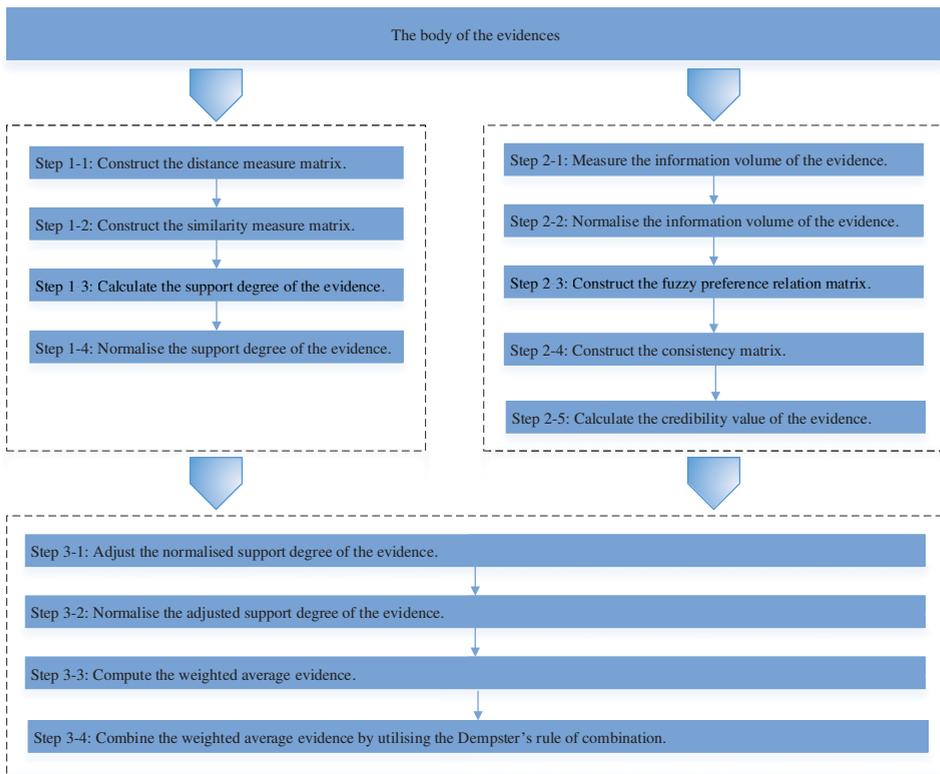


Figure 1. The flowchart of the proposed method.

3.1. Calculate the Support Degree of the Evidence

Step 1: The distance measure d_{ij} between the BPAs m_i ($i = 1, 2, \dots, k$) and m_j ($j = 1, 2, \dots, k$) can be obtained by Equations (9) and (10); thus, a distance measure matrix $DMM = (d_{ij})_{k \times k}$ can be constructed as follows:

$$DMM = \begin{bmatrix} 0 & \cdots & d_{1i} & \cdots & d_{1k} \\ \vdots & \cdots & \vdots & \vdots & \vdots \\ d_{i1} & \cdots & 0 & \cdots & d_{ik} \\ \vdots & \cdots & \vdots & \vdots & \vdots \\ d_{k1} & \cdots & d_{ki} & \cdots & 0 \end{bmatrix}. \tag{18}$$

Step 2: The similarity measure S_{ij} between the BPAs m_i and m_j can be obtained by:

$$S_{ij} = 1 - d_{ij}, \quad 1 \leq i \leq k; 1 \leq j \leq k. \tag{19}$$

Then, the similarity measure matrix $SMM = (S_{ij})_{k \times k}$ can be constructed as follows:

$$SMM = \begin{bmatrix} 1 & \cdots & S_{1i} & \cdots & S_{1k} \\ \vdots & \cdots & \vdots & \vdots & \vdots \\ S_{i1} & \cdots & 1 & \cdots & S_{ik} \\ \vdots & \cdots & \vdots & \vdots & \vdots \\ S_{k1} & \cdots & S_{ki} & \cdots & 1 \end{bmatrix}. \tag{20}$$

Step 3: The support degree of the BPA m_i is defined as follows:

$$Sup_i = \sum_{j=1, j \neq i}^k S_{ij}, \quad 1 \leq i \leq k. \tag{21}$$

Step 4: The support degree of the BPA m_i is normalized as below, which is denoted as \tilde{Sup}_i :

$$\tilde{Sup}_i = \frac{Sup_i}{\sum_{r=1}^k Sup_r}, \quad 1 \leq i \leq k. \tag{22}$$

3.2. Generate the Credibility Value of the Evidence

In the course of information fusion, it is important to identify the relatively credible evidence in terms of the obtained pieces of evidence. Due to the increase of the uncertainty in the collection of information, the degree of anarchy involved in the systems rises, which violates the necessary condition to use Dempster’s rule of combination. Utilizing the ordered information can make the technologies based on the Dempster–Shafer evidence theory more robust. Therefore, we take advantage of the fuzzy preference relations analysis [79] based on the belief entropy [43] to indicate the relative credibility preference among the pieces of evidence. The concrete steps are listed as follows:

Step 1: The belief entropy of the BPA m_i ($i = 1, 2, \dots, k$) is calculated by leveraging Equation (11).

Because the belief entropy of the evidence may be zero in a certain case, in order to avoid allocating zero weight to such kinds of evidence, we utilize the information volume IV_i for measuring the uncertainty of the BPA m_i as below:

$$IV_i = e^{E_d(m_i)} = e^{-\sum_{A \subseteq \Theta} m(A) \log \frac{m(A)}{2^{|A|-1}}}, \quad 1 \leq i \leq k. \tag{23}$$

Step 2: The information volume of the BPA m_i is normalized as below, which is denoted as \tilde{IV}_i :

$$\tilde{IV}_i = \frac{IV_i}{\sum_{r=1}^k IV_r}, \quad 1 \leq i \leq k. \tag{24}$$

Step 3: The fuzzy preference relation matrix $P = (p_{ij})_{k \times k}$, where $p_{ij} \in [0, 1]$ can be constructed by the following steps:

- Step 3-1: According to Definition 6, the diagonal element p_{ii} is assigned to 0.5.
- Step 3-2: If there are only two pieces of evidence, all of the off-diagonal elements p_{ij} and p_{ji} will be assigned to 0.5, because we have no sufficient evidence to detect how the pieces of evidence are preferred with respect to each other. Thus, the fuzzy preference relation matrix $P = (p_{ij})_{k \times k}$ can be constructed by:

$$P = (p_{ij})_{k \times k} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}. \tag{25}$$

Step 3-3: If there are more than two pieces of evidence, the variance of entropy for the BPA m_i ($1 \leq i \leq k$) will be calculated as follows:

$$Var_i = Var(\{\tilde{I}V_1, \tilde{I}V_2, \dots, \tilde{I}V_{i-1}, \tilde{I}V_{i+1}, \dots, \tilde{I}V_k\}). \tag{26}$$

Step 3-4: The smaller the value Var_i has, the more conflict the evidence has in the decision-making system, so that a small preference value is supposed to be assigned to this evidence. Otherwise, the bigger the value Var_i has, the less conflict the evidence has in the decision-making system, so that a big preference value is supposed to be assigned to this evidence. On the basis of the above variance of entropy, the off-diagonal elements p_{ij} and p_{ji} will be computed by Equations (27) and (28) introduced in [79].

$$p_{ij} = \frac{Var_i}{Var_i + Var_j}, \tag{27}$$

$$p_{ji} = \frac{Var_j}{Var_i + Var_j}, \tag{28}$$

where $1 \leq i \leq k$ and $1 \leq j \leq k$.

Step 4: Based on the obtained fuzzy preference relation matrix $P = (p_{ij})_{k \times k}$, the consistency matrix \bar{P} can be constructed by Equation (16).

Step 5: With the consistency matrix \bar{P} , the credibility value of the BPA m_i is defined based on Equation (17):

$$Crd_i = \frac{2}{k^2} \sum_{j=1}^k \bar{p}_{ij}, \quad 1 \leq i \leq k. \tag{29}$$

We can notice that $\sum_{i=1}^k Crd_i = 1$. Hence, the credibility value of each piece of evidence is regarded as a weight that indicates the relative credibility preference in terms of the evidence.

3.3. Fuse the Weighted Average Evidence

Step 1: Based on the credibility degree Crd_i , the normalized support degree of the BPA m_i will be adjusted, denoted as $ASup_i$:

$$ASup_i = Crd_i \times \tilde{Sup}_i, \quad 1 \leq i \leq k. \tag{30}$$

Step 2: The $ASup_i$ is normalized as below, denoted as \tilde{ASup}_i , which is considered as the final weight of the BPA m_i .

$$\tilde{ASup}_i = \frac{ASup_i}{\sum_{r=1}^k ASup_r}, \quad 1 \leq i \leq k. \tag{31}$$

Step 3: On the basis of the final weight \tilde{ASup}_i , the weighted average evidence $WAE(m)$ can be obtained as follows:

$$WAE(m) = \sum_{i=1}^k (\tilde{ASup}_i \times m_i), \quad 1 \leq i \leq k, \tag{32}$$

where k denotes the number of BPAs and m_i represents the i -th BPA, which are modeled from the sensor reports.

Step 4: The weighted average evidence $WAE(m)$ is combined through Dempster’s combination rule, namely Equation (7), by $k - 1$ times, if there are k number of pieces of evidence. Then, the final combination result of multiple pieces of evidence can be obtained.

4. Experiment

In this section, to demonstrate the effectiveness of the proposal, a numerical example is illustrated.

Example 1. Consider a target recognition problem based on multiple sensors associated with the sensor reports that are collected from five different types of sensors. These sensor reports that are modeled as the BPAs are given in Table 1 from [72], where the frame of discernment Θ that consists of three potential objects is given by $\Theta = \{A, B, C\}$.

Table 1. The basic probability assignments (BPAs) for Example 1.

BPA	{A}	{B}	{C}	{A, C}
$S_1 : m_1(\cdot)$	0.41	0.29	0.30	0.00
$S_2 : m_2(\cdot)$	0.00	0.90	0.10	0.00
$S_3 : m_3(\cdot)$	0.58	0.07	0.00	0.35
$S_4 : m_4(\cdot)$	0.55	0.10	0.00	0.35
$S_5 : m_5(\cdot)$	0.60	0.10	0.00	0.30

Step 1: Construct the distance measure matrix $DMM = (d_{ij})_{k \times k}$ as follows:

$$DMM = \begin{pmatrix} 0 & 0.5386 & 0.3495 & 0.3257 & 0.3311 \\ 0.5386 & 0 & 0.8142 & 0.7850 & 0.7906 \\ 0.3495 & 0.8142 & 0 & 0.0300 & 0.0374 \\ 0.3257 & 0.7850 & 0.0300 & 0 & 0.0354 \\ 0.3311 & 0.7906 & 0.0374 & 0.0354 & 0 \end{pmatrix}.$$

Step 2: Construct the similarity measure matrix $SMM = (S_{ij})_{k \times k}$ as follows:

$$SMM = \begin{pmatrix} 1 & 0.4614 & 0.6505 & 0.6743 & 0.6689 \\ 0.4614 & 1 & 0.1858 & 0.2150 & 0.2094 \\ 0.6505 & 0.1858 & 1 & 0.9700 & 0.9626 \\ 0.6743 & 0.2150 & 0.9700 & 1 & 0.9646 \\ 0.6689 & 0.2094 & 0.9626 & 0.9646 & 1 \end{pmatrix}.$$

Step 3: Calculate the support degree of the BPA m_i as below:

$$\begin{aligned} Sup_1 &= 2.4551, \\ Sup_2 &= 1.0716, \\ Sup_3 &= 2.7689, \\ Sup_4 &= 2.8239, \\ Sup_5 &= 2.8055. \end{aligned}$$

Step 4: Normalize the support degree of the BPA m_i as follows:

$$\begin{aligned} \tilde{Sup}_1 &= 0.2059, \\ \tilde{Sup}_2 &= 0.0899, \\ \tilde{Sup}_3 &= 0.2322, \\ \tilde{Sup}_4 &= 0.2368, \\ \tilde{Sup}_5 &= 0.2353. \end{aligned}$$

Step 5: Measure the information volume of the BPA m_i as below:

$$\begin{aligned} IV_1 &= 4.7894, \\ IV_2 &= 1.5984, \\ IV_3 &= 6.1056, \\ IV_4 &= 6.6286, \\ IV_5 &= 5.8767. \end{aligned}$$

Step 6: Normalize the information volume of the BPA m_i as follows:

$$\begin{aligned} \tilde{IV}_1 &= 0.1916, \\ \tilde{IV}_2 &= 0.0639, \end{aligned}$$

$$\begin{aligned}\tilde{IV}_3 &= 0.2442, \\ \tilde{IV}_4 &= 0.2652, \\ \tilde{IV}_5 &= 0.2351.\end{aligned}$$

Step 7: Construct the fuzzy preference relation matrix $P = (p_{ij})_{n \times n}$ as follows:

$$P = \begin{pmatrix} 0.5000 & 0.9002 & 0.5238 & 0.5559 & 0.5144 \\ 0.0998 & 0.5000 & 0.1087 & 0.1219 & 0.1051 \\ 0.4762 & 0.8913 & 0.5000 & 0.5323 & 0.4906 \\ 0.4441 & 0.8781 & 0.4677 & 0.5000 & 0.4583 \\ 0.4856 & 0.8949 & 0.5094 & 0.5417 & 0.5000 \end{pmatrix}.$$

Step 8: Construct the consistency matrix $\bar{P} = (\bar{p}_{ik})_{n \times n}$ as follows:

$$\bar{P} = \begin{pmatrix} 0.5000 & 0.9117 & 0.5208 & 0.5492 & 0.5125 \\ 0.0883 & 0.5000 & 0.1091 & 0.1375 & 0.1008 \\ 0.4792 & 0.8909 & 0.5000 & 0.5284 & 0.4917 \\ 0.4508 & 0.8625 & 0.4716 & 0.5000 & 0.4633 \\ 0.4875 & 0.8992 & 0.5083 & 0.5367 & 0.5000 \end{pmatrix}.$$

Step 9: Calculate the credibility value of the BPA m_i as below:

$$\begin{aligned}Crd_1 &= 0.2395, \\ Crd_2 &= 0.0749, \\ Crd_3 &= 0.2312, \\ Crd_4 &= 0.2198, \\ Crd_5 &= 0.2345.\end{aligned}$$

Step 10: Adjust the normalized support degree of the BPA m_i based on the credibility value as below:

$$\begin{aligned}ASup_1 &= 0.0493, \\ ASup_2 &= 0.0067, \\ ASup_3 &= 0.0537, \\ ASup_4 &= 0.0521, \\ ASup_5 &= 0.0552.\end{aligned}$$

Step 11: Normalize the adjusted support degree of the BPA m_i as below:

$$\begin{aligned}\tilde{ASup}_1 &= 0.2273, \\ \tilde{ASup}_2 &= 0.0310, \\ \tilde{ASup}_3 &= 0.2474, \\ \tilde{ASup}_4 &= 0.2399, \\ \tilde{ASup}_5 &= 0.2543.\end{aligned}$$

Step 12: Compute the weighted average evidence as below:

$$\begin{aligned}m(\{A\}) &= 0.5213, \\ m(\{B\}) &= 0.1606, \\ m(\{C\}) &= 0.0713, \\ m(\{A, C\}) &= 0.2469.\end{aligned}$$

Step 13: Combine the weighted average evidence by utilizing Dempster's rule of combination four times. The results of the combination for the first time are shown below:

$$\begin{aligned}m(\{A\}) &= 0.8066, \\ m(\{B\}) &= 0.0393, \\ m(\{C\}) &= 0.0614, \\ m(\{A, C\}) &= 0.0929.\end{aligned}$$

For the combination for the second time, the results are listed as follows:

$$\begin{aligned}m(\{A\}) &= 0.9239, \\m(\{B\}) &= 0.0087, \\m(\{C\}) &= 0.0362, \\m(\{A, C\}) &= 0.0317.\end{aligned}$$

Next, the results of the third combination are calculated as:

$$\begin{aligned}m(\{A\}) &= 0.9701, \\m(\{B\}) &= 0.0019, \\m(\{C\}) &= 0.0184, \\m(\{A, C\}) &= 0.0105.\end{aligned}$$

Then, the combination results of the fourth time, namely the final fusing results, are produced as follows:

$$\begin{aligned}m(\{A\}) &= 0.9888, \\m(\{B\}) &= 0.0004, \\m(\{C\}) &= 0.0087, \\m(\{A, C\}) &= 0.0034.\end{aligned}$$

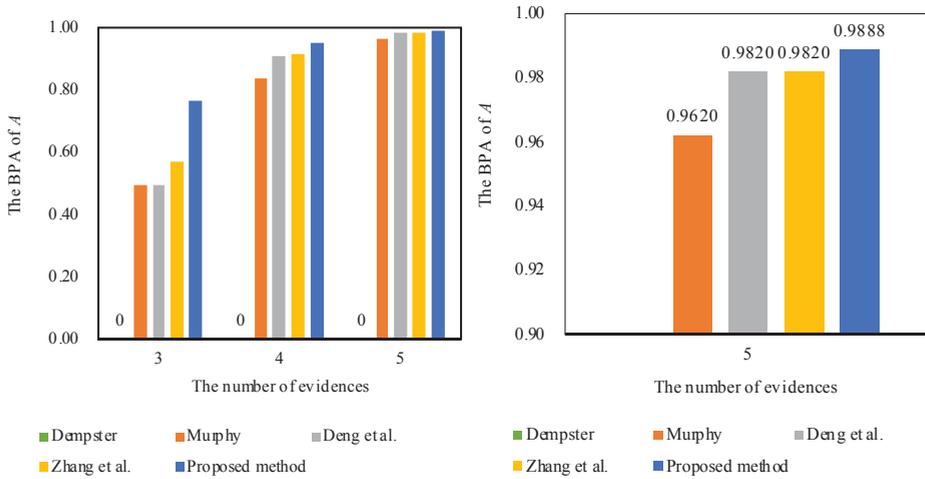
From Example 1, we can notice that the evidence m_2 highly conflicts with other pieces of evidence, because the normalized support degree of the evidence m_2 is 0.0899, which is much lower than the normalized support degrees of other pieces of evidence.

The fusing results that are generated by different combination methods are shown in Table 2. The comparisons of the BPA of the target A by different combination rules are shown in Figure 2.

Table 2. Combination results of the evidence in terms of different combination rules.

Evidence	Method	{A}	{B}	{C}	{AC}	Target
m_1, m_2, m_3	Dempster [23]	0	0.6350	0.3650	0	B
	Murphy [71]	0.4939	0.4180	0.0792	0.0090	A
	Deng et al. [72]	0.4974	0.4054	0.0888	0.0084	A
	Zhang et al. [73]	0.5681	0.3319	0.0929	0.0084	A
	Proposed method	0.7617	0.1127	0.1176	0.0080	A
m_1, m_2, m_3, m_4	Dempster [23]	0	0.3321	0.6679	0	C
	Murphy [71]	0.8362	0.1147	0.0410	0.0081	A
	Deng et al. [72]	0.9089	0.0444	0.0379	0.0089	A
	Zhang et al. [73]	0.9142	0.0395	0.0399	0.0083	A
	Proposed method	0.9507	0.0060	0.0334	0.0087	A
m_1, m_2, m_3, m_4, m_5	Dempster [23]	0	0.1422	0.8578	0	C
	Murphy [71]	0.9620	0.0210	0.0138	0.0032	A
	Deng et al. [72]	0.9820	0.0039	0.0107	0.0034	A
	Zhang et al. [73]	0.9820	0.0034	0.0115	0.0032	A
	Proposed method	0.9888	0.0004	0.0087	0.0034	A

As shown in Table 2, Dempster's combination rule generates a counterintuitive result, even though the other four pieces of evidence support the target A . As the number of pieces of evidence increases from 3–5, Murphy's method [71], Deng et al.'s method [72], Zhang et al.'s method [73] and the proposed method present reasonable results. Additionally, the proposed method is efficient in dealing with the conflicting pieces of evidence with better convergence as shown in Figure 2. The reason is that the proposal not only makes use of the function of evidence distance to obtain the evidence's support degree, but also adopts the fuzzy preference relations analysis based on the belief entropy to measure the relative credibility preference among the pieces of evidence. After considering these aspects, the unreliable evidence's weight is decreased, so that its negative effect can be relieved on the final fusing results compared to other methods.



(a)The comparison of the BPA of the target A when the number of evidence increases from 3 to 5. (b)The comparison of the BPA of the target A when the number of evidence is 5.

Figure 2. The comparison of different methods in Example 1.

5. Application

In this section, the proposal is applied to the fault diagnosis of a motor rotor, where the practical data in [27] are used for the comparison with the related method.

5.1. Problem Statement

Supposing that the frame of discernment Θ , which consists of three types of faults for a motor rotor is given by $\Theta = \{Rotor\ unbalance, Rotor\ misalignment, Pedestal\ looseness\} = \{F_1, F_2, F_3\}$. The set of vibration acceleration sensors given by $S = \{S_1, S_2, S_3\}$ is positioned at different places for gathering the vibration signals. The acceleration vibration frequency amplitudes at 1X frequency, 2X frequency and 3X frequency are considered as the fault feature variables. The collected sensor reports at 1X frequency, 2X frequency and 3X frequency that are modeled as BPAs are given in Tables 3–5, respectively, where $m_1(\cdot), m_2(\cdot)$ and $m_3(\cdot)$ represent the BPAs reported from the three vibration acceleration sensors S_1, S_2 and S_3 .

Table 3. The collected sensor reports at the frequency of 1X modeled as BPAs.

BPA	$\{F_2\}$	$\{F_3\}$	$\{F_1, F_2\}$	$\{F_1, F_2, F_3\}$
$S_1 : m_1(\cdot)$	0.8176	0.0003	0.1553	0.0268
$S_2 : m_2(\cdot)$	0.5658	0.0009	0.0646	0.3687
$S_3 : m_3(\cdot)$	0.2403	0.0004	0.0141	0.7452

Table 4. The collected sensor reports at the frequency of 2X modeled as BPAs.

BPA	$\{F_2\}$	$\{F_1, F_2, F_3\}$
$S_1 : m_1(\cdot)$	0.6229	0.3771
$S_2 : m_2(\cdot)$	0.7660	0.2341
$S_3 : m_3(\cdot)$	0.8598	0.1402

Table 5. The collected sensor reports at the frequency of 3X modeled as BPAs.

BPA	$\{F_1\}$	$\{F_2\}$	$\{F_1, F_2\}$	$\{F_1, F_2, F_3\}$
$S_1 : m_1(\cdot)$	0.3666	0.4563	0.1185	0.0586
$S_2 : m_2(\cdot)$	0.2793	0.4151	0.2652	0.0404
$S_3 : m_3(\cdot)$	0.2897	0.4331	0.2470	0.0302

5.2. Motor Rotor Fault Diagnosis Based on the Proposed Method

5.2.1. Motor Rotor Fault Diagnosis at 1X Frequency

According to the proposed method in Section 3 and Table 3's BPAs modeled by the collected sensor reports at the frequency of 1X, the weighted average evidence in terms of motor rotor fault diagnosis at 1X frequency is obtained as follows:

$$\begin{aligned} m(\{F_2\}) &= 0.5636, \\ m(\{F_3\}) &= 0.0006, \\ m(\{F_1, F_2\}) &= 0.0782, \\ m(\{F_1, F_2, F_3\}) &= 0.3576. \end{aligned}$$

After that, the weighted average evidence in terms of motor rotor fault diagnosis at 1X frequency is fused by utilizing Dempster's rule of combination two times. The results of the combination for the first time are shown below:

$$\begin{aligned} m(\{F_2\}) &= 0.8095, \\ m(\{F_3\}) &= 0.0004, \\ m(\{F_1, F_2\}) &= 0.0621, \\ m(\{F_1, F_2, F_3\}) &= 0.1280. \end{aligned}$$

Then, the results of the combination for the second time, namely the final fusing results for motor rotor fault diagnosis at 1X frequency, are generated as follows:

$$\begin{aligned} m(\{F_2\}) &= 0.9169, \\ m(\{F_3\}) &= 0.0002, \\ m(\{F_1, F_2\}) &= 0.0371, \\ m(\{F_1, F_2, F_3\}) &= 0.0458. \end{aligned}$$

5.2.2. Motor Rotor Fault Diagnosis at 2X Frequency

On the basis of the proposed method in Section 3 and Table 4's BPAs modeled by the collected sensor reports at the frequency of 2X, the weighted average evidence with respect to motor rotor fault diagnosis at 2X frequency is obtained as follows:

$$\begin{aligned} m(\{F_2\}) &= 0.7754, \\ m(\{F_1, F_2, F_3\}) &= 0.2246. \end{aligned}$$

Next, by leveraging Dempster's rule of combination, the weighted average evidence with respect to motor rotor fault diagnosis at 2X frequency is fused two times. For the combination for the first time, the fusion results are given below:

$$\begin{aligned} m(\{F_2\}) &= 0.9496, \\ m(\{F_1, F_2, F_3\}) &= 0.0504. \end{aligned}$$

Afterwards, for the combination for the second time, the final fusion results with respect to motor rotor fault diagnosis at 2X frequency are shown below:

$$m(\{F_2\}) = 0.9887,$$

$$m(\{F_1, F_2, F_3\}) = 0.0113.$$

5.2.3. Motor Rotor Fault Diagnosis at 3X Frequency

By applying the proposed method in Section 3 and Table 5's BPAs modeled by the collected sensor reports at the frequency of 3X, the weighted average evidence with regard to motor rotor fault diagnosis at 3X frequency is obtained as follows:

$$\begin{aligned} m(\{F_1\}) &= 0.3028, \\ m(\{F_2\}) &= 0.4323, \\ m(\{F_1, F_2\}) &= 0.2254, \\ m(\{F_1, F_2, F_3\}) &= 0.0395. \end{aligned}$$

Therewith, the weighted average evidence with regard to motor rotor fault diagnosis at 3X frequency is fused by utilizing Dempster's rule of combination two times. The combination results for the first time are listed below:

$$\begin{aligned} m(\{F_1\}) &= 0.3415, \\ m(\{F_2\}) &= 0.5634, \\ m(\{F_1, F_2\}) &= 0.0929, \\ m(\{F_1, F_2, F_3\}) &= 0.0021. \end{aligned}$$

Then, the final combination results for the second time for motor rotor fault diagnosis at 3X frequency are shown below:

$$\begin{aligned} m(\{F_1\}) &= 0.3266, \\ m(\{F_2\}) &= 0.6365, \\ m(\{F_1, F_2\}) &= 0.0368, \\ m(\{F_1, F_2, F_3\}) &= 0.0001. \end{aligned}$$

5.3. Discussion

According to the results as shown in Tables 6–8, we can notice that the proposed method can diagnose the fault type F_2 , which is consistent with Jiang et al.'s method [27]. Even facing the conflicting sensor reports where the normalized support degrees of the sensor reports are different at 1X frequency, 2X frequency and 3X frequency, both of the methods can well manage the conflicting pieces of evidence and diagnose the fault type F_2 .

Furthermore, the proposed method outperforms Jiang et al.'s method [27] in terms of dealing with the conflicting pieces of evidence, as well as coping with the uncertainty as shown in Figures 3–5, because the belief degrees assigned to the target F_2 at 1X frequency, 2X frequency and 3X frequency by the proposed method rise to 91.69%, 98.87% and 63.65%, respectively, while the belief degrees assigned to the target F_2 at 1X frequency, 2X frequency and 3X frequency by the method Jiang et al. [27] are 88.61%, 96.21% and 59.04%, respectively.

On the other hand, the uncertainty $\{F_1, F_2\}$ falls to 0.0371 from 0.0582, and the uncertainty $\{F_1, F_2, F_3\}$ falls to 0.0458 from 0.0555 at 1X frequency; the uncertainty $\{F_1, F_2, F_3\}$ drops to 0.0113 from 0.0371 at 2X frequency; the uncertainty $\{F_1, F_2\}$ falls to 0.0368 from 0.0651, and the uncertainty $\{F_1, F_2, F_3\}$ falls to 0.0001 from 0.0061 at 3X frequency. The main reason is that the proposed method not only takes the support degree of the sensor reports into account by making use of the function of evidence distance, but also considers the relative credibility preference of the sensor reports by taking advantage of the fuzzy preference relations analysis on the basis of the belief entropy. As a result, the proposed method can diagnose motor rotor fault more accurately.

Table 6. Fusion results of different methods for motor rotor fault diagnosis at 1X frequency.

Method	$\{F_2\}$	$\{F_3\}$	$\{F_1, F_2\}$	$\{F_1, F_2, F_3\}$	Target
Jiang et al. [27]	0.8861	0.0002	0.0582	0.0555	F_2
Proposed method	0.9169	0.0002	0.0371	0.0458	F_2

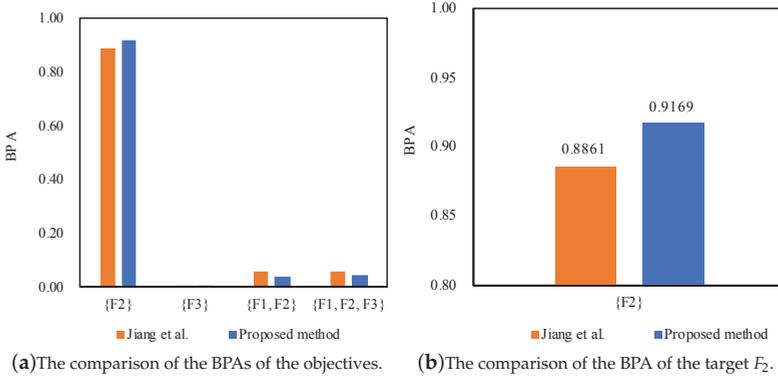


Figure 3. The comparison of different methods for motor rotor fault diagnosis at 1X frequency.

Table 7. Fusion results of different methods for motor rotor fault diagnosis at 2X frequency.

Method	$\{F_2\}$	$\{F_1, F_2, F_3\}$	Target
Jiang et al. [27]	0.9621	0.0371	F_2
Proposed method	0.9887	0.0113	F_2

Table 8. Fusion results of different methods for motor rotor fault diagnosis at 3X frequency.

Method	$\{F_1\}$	$\{F_2\}$	$\{F_1, F_2\}$	$\{F_1, F_2, F_3\}$	Target
Jiang et al. [27]	0.3384	0.5904	0.0651	0.0061	F_2
Proposed method	0.3266	0.6365	0.0368	0.0001	F_2

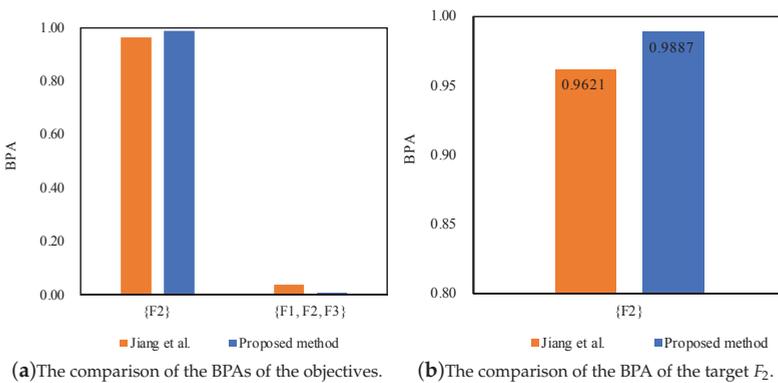


Figure 4. The comparison of different methods for motor rotor fault diagnosis at 2X frequency.

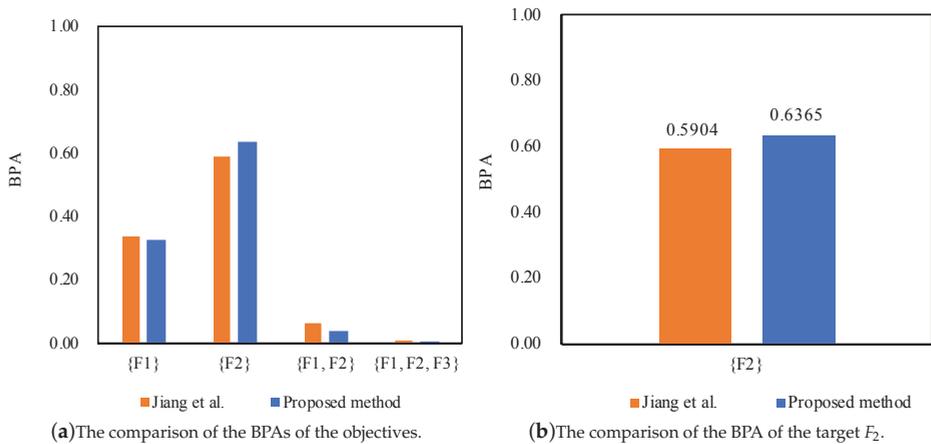


Figure 5. The comparison of different methods for motor rotor fault diagnosis at 3X frequency.

6. Conclusions

In this paper, on account of the support degree among the pieces of evidence, the uncertainty measure of the evidence and the effect of the relative credibility of evidence on the weight, a novel method for multi-sensor data fusion was proposed. The proposed method was a hybrid methodology by integrating the distance of evidence, belief entropy and fuzzy preference relation analysis. It consisted of three main procedures. Firstly, the support degree of the evidence was calculated to represent the reliability of the evidence. Secondly, the credibility value of the evidence was generated to indicate the relative credibility preference of the evidence. Thirdly, based on the first two procedures, the weighted average evidence was obtained; thus, it could be fused by applying Dempster's combination rule. As described above, the proposed method was a kind of approach to pre-process the bodies of evidence. Through a numerical example, it was illustrated that the proposal was more effective and feasible than other related methods to handle the conflicting evidence combination problem under a multi-sensor environment with better convergence. On the other hand, a practical application in fault diagnosis was presented to demonstrate that the proposed method could diagnose the faults more accurately.

In future work, I intend to consider further fault diagnosis of complicated equipment/systems that involves certain faults, such as cracks and misalignment. On the other hand, multiple faults, like bearing faults, rotor-related faults, etc., will be taken into account in future work to improve the robustness of the technique.

Acknowledgments: This research is supported by the Fundamental Research Funds for the Central Universities (Grant No. SWU115008), the National Natural Science Foundation of China (Nos. 61672435, 61702427, 61702426) and the 1000-Plan of Chongqing by Southwest University (No. SWU116007).

Author Contributions: Fuyuan Xiao contributed to all the work in this paper.

Conflicts of Interest: The author declares no conflict of interest.

References

- Hall, D.L.; Llinas, J. An introduction to multisensor data fusion. *Proc. IEEE* **1997**, *85*, 6–23.
- Zois, E.N.; Anastassopoulos, V. Fusion of correlated decisions for writer verification. *Pattern Recognit.* **2001**, *34*, 47–61.
- Dong, M.; He, D. Hidden semi-Markov model-based methodology for multi-sensor equipment health diagnosis and prognosis. *Eur. J. Oper. Res.* **2007**, *178*, 858–878.

4. Pohl, V. Multisensor image fusion in remote sensing: Concepts, methods and applications. *Int. J. Remote Sens.* **1998**, *19*, 823–854.
5. Dong, J.; Zhuang, D.; Huang, Y.; Fu, J. Advances in multi-sensor data fusion: Algorithms and applications. *Sensors* **2009**, *9*, 7771–7784.
6. Yang, F.; Wei, H. Fusion of infrared polarization and intensity images using support value transform and fuzzy combination rules. *Infrared Phys. Technol.* **2013**, *60*, 235–243.
7. Shen, X.; Varshney, P.K. Sensor selection based on generalized information gain for target tracking in large sensor networks. *IEEE Trans. Signal Process.* **2014**, *62*, 363–375.
8. Jiang, W.; Wei, B.; Xie, C.; Zhou, D. An evidential sensor fusion method in fault diagnosis. *Adv. Mech. Eng.* **2016**, *8*, doi:10.1177/1687814016641820.
9. Deng, X.; Jiang, W.; Zhang, J. Zero-sum matrix game with payoffs of Dempster–Shafer belief structures and its applications on sensors. *Sensors* **2017**, *17*, 922.
10. Yunusa-Kaltungo, A.; Sinha, J.K.; Elbhah, K. An improved data fusion technique for faults diagnosis in rotating machines. *Measurement* **2014**, *58*, 27–32.
11. Liu, Q.C.; Wang, H.P.B. A case study on multisensor data fusion for imbalance diagnosis of rotating machinery. *Artif. Intell. Eng. Des. Anal. Manuf.* **2001**, *15*, 203–210.
12. Safizadeh, M.; Latifi, S. Using multi-sensor data fusion for vibration fault diagnosis of rolling element bearings by accelerometer and load cell. *Inf. Fusion* **2014**, *18*, 1–8.
13. Banerjee, T.P.; Das, S. Multi-sensor data fusion using support vector machine for motor fault detection. *Inf. Sci.* **2012**, *217*, 96–107.
14. Niu, G.; Han, T.; Yang, B.S.; Tan, A.C.C. Multi-agent decision fusion for motor fault diagnosis. *Mech. Syst. Signal Process.* **2007**, *21*, 1285–1299.
15. Yunusa-Kaltungo, A.; Sinha, J.K. Sensitivity analysis of higher order coherent spectra in machine faults diagnosis. *Struct. Health Monit. Int. J.* **2016**, *15*, 555–567.
16. Walczak, B.; Massart, D. Rough sets theory. *Chemom. Intell. Lab. Syst.* **1999**, *47*, 1–16.
17. Shen, L.; Tay, F.E.; Qu, L.; Shen, Y. Fault diagnosis using rough sets theory. *Comput. Ind.* **2000**, *43*, 61–72.
18. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353.
19. Chen, S.; Deng, Y.; Wu, J. Fuzzy sensor fusion based on evidence theory and its application. *Appl. Artif. Intell.* **2013**, *27*, 235–248.
20. Liu, H.C.; Liu, L.; Lin, Q.L. Fuzzy failure mode and effects analysis using fuzzy evidential reasoning and belief rule-based methodology. *IEEE Trans. Reliab.* **2013**, *62*, 23–36.
21. Mardani, A.; Jusoh, A.; Zavadskas, E.K. Fuzzy multiple criteria decision-making techniques and applications—Two decades review from 1994 to 2014. *Expert Syst. Appl.* **2015**, *42*, 4126–4148.
22. Zheng, H.; Deng, Y.; Hu, Y. Fuzzy evidential influence diagram and its evaluation algorithm. *Knowl.-Based Syst.* **2017**, *131*, 28–45.
23. Dempster, A.P. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339.
24. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976; Volume 1.
25. Jiang, W.; Zhan, J. A modified combination rule in generalized evidence theory. *Appl. Intell.* **2017**, *46*, 630–640.
26. Zadeh, L.A. A note on Z-numbers. *Inf. Sci.* **2011**, *181*, 2923–2932.
27. Jiang, W.; Xie, C.; Zhuang, M.; Shou, Y.; Tang, Y. Sensor data fusion with Z-numbers and its application in fault diagnosis. *Sensors* **2016**, *16*, 1509.
28. Xiao, F. An intelligent complex event processing with D numbers under fuzzy environment. *Math. Probl. Eng.* **2016**, *2016*, 3713518.
29. Mo, H.; Deng, Y. A new aggregating operator for linguistic information based on D numbers. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2016**, *24*, 831–846.
30. Zhou, X.; Deng, X.; Deng, Y.; Mahadevan, S. Dependence assessment in human reliability analysis based on D numbers and AHP. *Nucl. Eng. Des.* **2017**, *313*, 243–252.
31. Yang, J.B.; Wang, Y.M.; Xu, D.L.; Chin, K.S.; Chatton, L. Belief rule-based methodology for mapping consumer preferences and setting product targets. *Expert Syst. Appl.* **2012**, *39*, 4749–4759.
32. Fu, C.; Yang, J.B.; Yang, S.L. A group evidential reasoning approach based on expert reliability. *Eur. J. Oper. Res.* **2015**, *246*, 886–893.

33. Yang, J.B.; Xu, D.L. Interactive minimax optimisation for integrated performance analysis and resource planning. *Comput. Oper. Res.* **2014**, *46*, 78–90.
34. Yang, J.B.; Xu, D.L. Evidential reasoning rule for evidence combination. *Artif. Intell.* **2013**, *205*, 1–29.
35. Ma, J. Qualitative approach to Bayesian networks with multiple causes. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2012**, *42*, 382–391.
36. Graziani, S.; Xibilia, M. A deep learning based soft sensor for a sour water stripping plant. In Proceedings of the 2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Turin, Italy, 22–25 May 2017; pp. 1–6.
37. Xu, S.; Jiang, W.; Deng, X.; Shou, Y. A modified Physarum-inspired model for the user equilibrium traffic assignment problem. *Appl. Math. Model.* **2017**, doi:10.1016/j.apm.2017.07.032.
38. Xibilia, M.; Gemelli, N.; Consolo, G. Input variables selection criteria for data-driven Soft Sensors design. In Proceedings of the 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), Calabria, Italy, 16–18 May 2017; pp. 362–367.
39. Denoeux, T. A k -nearest neighbor classification rule based on Dempster–Shafer theory. *IEEE Trans. Syst. Man Cybernet.* **1995**, *25*, 804–813.
40. Ma, J.; Liu, W.; Miller, P.; Zhou, H. An evidential fusion approach for gender profiling. *Inf. Sci.* **2016**, *333*, 10–20.
41. Liu, Z.-G.; Pan, Q.; Dezert, J.; Martin, A. Adaptive imputation of missing values for incomplete pattern classification. *Pattern Recognit.* **2016**, *52*, 85–95.
42. Fu, C.; Xu, D.L. Determining attribute weights to improve solution reliability and its application to selecting leading industries. *Ann. Oper. Res.* **2016**, *245*, 401–426.
43. Deng, Y. Deng entropy. *Chaos Solitons Fractals* **2016**, *91*, 549–553.
44. Deng, X.; Jiang, W. An evidential axiomatic design approach for decision making using the evaluation of belief structure satisfaction to uncertain target values. *Int. J. Intell. Syst.* **2017**, doi:10.1002/int.21929.
45. Rikhtegar, N.; Mansouri, N.; Ahadi Oroumieh, A.; Yazdani-Chamzini, A.; Kazimieras Zavadskas, E.; Kildienė, S. Environmental impact assessment based on group decision-making methods in mining projects. *Econ. Res.-Ekonomika Istraživanja* **2014**, *27*, 378–392.
46. Jiang, W.; Wang, S.; Liu, X.; Zheng, H.; Wei, B. Evidence conflict measure based on OWA operator in open world. *PLoS ONE* **2017**, *12*, e0177828.
47. Jiang, W.; Wang, S. An uncertainty measure for interval-valued evidences. *Int. J. Comput. Commun. Control* **2017**, *12*, 631–644.
48. Zhang, X.; Deng, Y.; Chan, F.T.S.; Adamatzky, A.; Mahadevan, S. Supplier selection based on evidence theory and analytic network process. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2016**, *230*, 562–573.
49. Liu, T.; Deng, Y.; Chan, F. Evidential supplier selection based on DEMATEL and game theory. *Int. J. Fuzzy Syst.* **2017**, doi:10.1007/s40815-017-0400-4.
50. Dong, Y.; Wang, J.; Chen, F.; Hu, Y.; Deng, Y. Location of Facility Based on Simulated Annealing and “ZKW” Algorithms. *Math. Probl. Eng.* **2017**, *2017*, 4628501.
51. Kang, B.; Chhipi-Shrestha, G.; Deng, Y.; Mori, J.; Hewage, K.; Sadiq, R. Development of a predictive model for Clostridium difficile infection incidence in hospitals using Gaussian mixture model and Dempster-Shafer theory. *Stoch. Environ. Res. Risk Assess.* **2017**, 1–16, doi:10.1007/s00477-017-1459-z.
52. Dutta, P. Uncertainty modeling in risk assessment based on Dempster–Shafer theory of evidence with generalized fuzzy focal elements. *Fuzzy Inf. Eng.* **2015**, *7*, 15–30.
53. Zheng, X.; Deng, Y. Dependence assessment in human reliability analysis based on evidence credibility decay model and IOWA operator. *Ann. Nucl. Energy* **2017**, in press.
54. Zhang, L.; Ding, L.; Wu, X.; Skibniewski, M.J. An improved Dempster–Shafer approach to construction safety risk perception. *Knowl.-Based Syst.* **2017**, *132*, 30–46.
55. Hang, J.; Zhang, J.; Cheng, M. Fault diagnosis of wind turbine based on multi-sensors information fusion technology. *IET Renew. Power Gener.* **2014**, *8*, 289–298.
56. Cheng, G.; Chen, X.-H.; Shan, X.-L.; Liu, H.-G.; Zhou, C.-F. A new method of gear fault diagnosis in strong noise based on multi-sensor information fusion. *J. Vib. Control* **2016**, *22*, 1504–1515.
57. Yuan, K.; Xiao, F.; Fei, L.; Kang, B.; Deng, Y. Modeling sensor reliability in fault diagnosis based on evidence theory. *Sensors* **2016**, *16*, 113.

58. Sabahi, F. A novel generalized belief structure comprising unprecisiated uncertainty applied to aphasia diagnosis. *J. Biomed. Inf.* **2016**, *62*, 66–77.
59. Du, Y.; Lu, X.; Su, X.; Hu, Y.; Deng, Y. New failure mode and effects analysis: An evidential downscaling method. *Qual. Reliab. Eng. Int.* **2016**, *32*, 737–746.
60. Jiang, W.; Xie, C.; Zhuang, M.; Tang, Y. Failure mode and effects analysis based on a novel fuzzy evidential method. *Appl. Soft Comput.* **2017**, *57*, 672–683.
61. Zadeh, L.A. A simple view of the Dempster–Shafer theory of evidence and its implication for the rule of combination. *AI Mag.* **1986**, *7*, 85–90.
62. Deng, Y. Generalized evidence theory. *Appl. Intell.* **2015**, *43*, 530–543.
63. Lefevre, E.; Colot, O.; Vannoorenberghe, P. Belief function combination and conflict management. *Inf. Fusion* **2002**, *3*, 149–162.
64. Jiang, W.; Wei, B.; Tang, Y.; Zhou, D. Ordered visibility graph average aggregation operator: An application in produced water management. *Chaos Interdiscip. J. Nonlinear Sci.* **2017**, *27*, 023117.
65. Han, D.; Deng, Y.; Han, C.Z.; Hou, Z. Weighted evidence combination based on distance of evidence and uncertainty measure. *J. Infrared Millim. Waves* **2011**, *30*, 396–400.
66. Deng, X.; Xiao, F.; Deng, Y. An improved distance-based total uncertainty measure in belief function theory. *Appl. Intell.* **2017**, *46*, 898–915.
67. Ma, J.; Liu, W.; Benferhat, S. A belief revision framework for revising epistemic states with partial epistemic states. *Int. J. Approx. Reason.* **2015**, *59*, 20–40.
68. Smets, P. The combination of evidence in the transferable belief model. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 447–458.
69. Dubois, D.; Prade, H. Representation and combination of uncertainty with belief functions and possibility measures. *Comput. Intell.* **1988**, *4*, 244–264.
70. Yager, R.R. On the Dempster–Shafer framework and new combination rules. *Inf. Sci.* **1987**, *41*, 93–137.
71. Murphy, C.K. Combining belief functions when evidence conflicts. *Decis. Support Syst.* **2000**, *29*, 1–9.
72. Deng, Y.; Shi, W.; Zhu, Z.; Liu, Q. Combining belief functions based on distance of evidence. *Decis. Support Syst.* **2004**, *38*, 489–493.
73. Zhang, Z.; Liu, T.; Chen, D.; Zhang, W. Novel algorithm for identifying and fusing conflicting data in wireless sensor networks. *Sensors* **2014**, *14*, 9562–9581.
74. Dempster, A.P. *A Generalization of Bayesian Inference*; Springer: Berlin/Heidelberg, Germany, 2008.
75. Jousselme, A.L.; Grenier, D.; Bossé, É. A new distance between two bodies of evidence. *Inf. Fusion* **2001**, *2*, 91–101.
76. Zhang, Q.; Li, M.; Deng, Y. Measure the structure similarity of nodes in complex networks based on relative entropy. *Phys. A Stat. Mech. Its Appl.* **2017**, doi:10.1016/j.physa.2017.09.042.
77. Shannon, C.E. A mathematical theory of communication. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2001**, *5*, 3–55.
78. Yager, R.R. Entropy and specificity in a mathematical theory of evidence. *Int. J. Gen. Syst.* **1983**, *9*, 249–260.
79. Tanino, T. Fuzzy preference orderings in group decision making. *Fuzzy Sets Syst.* **1984**, *12*, 117–131.
80. Lee, L.W. Group decision making with incomplete fuzzy preference relations based on the additive consistency and the order consistency. *Expert Syst. Appl.* **2012**, *39*, 11666–11676.
81. Fei, L.; Wang, H.; Chen, L.; Deng, Y. A new vector valued similarity measure for intuitionistic fuzzy sets based on OWA operators. *Iran. J. Fuzzy Syst.* **2017**, in press.
82. Fu, C.; Xu, D.L.; Yang, S.L. Distributed preference relations for multiple attribute decision analysis. *J. Oper. Res. Soc.* **2016**, *67*, 457–473.



Article

Centralized Multi-Sensor Square Root Cubature Joint Probabilistic Data Association

Yu Liu ^{1,2,*}, Jun Liu ^{2,*}, Gang Li ³, Lin Qi ², Yaowen Li ³ and You He ²¹ School of Electronic and Information Engineering, Beihang University, Beijing 100191, China² Research Institute of Information Fusion, Naval Aeronautical and Astronautical University, Yantai 264001, China; 3278pirate@163.com (L.Q.); heyou_f@126.com (Y.H.)³ Department of Electronic Engineering, Tsinghua University, Beijing 100084, China; gangli@mail.tsinghua.edu.cn (G.L.); lyw17@mails.tsinghua.edu.cn (Y.L.)

* Correspondence: liuyu77360132@126.com (Y.L.); 18615042187@163.com (J.L.); Tel.: +86-535-663-5877 (Y.L.)

Received: 8 September 2017; Accepted: 2 November 2017; Published: 5 November 2017

Abstract: This paper focuses on the tracking problem of multiple targets with multiple sensors in a nonlinear cluttered environment. To avoid Jacobian matrix computation and scaling parameter adjustment, improve numerical stability, and acquire more accurate estimated results for centralized nonlinear tracking, a novel centralized multi-sensor square root cubature joint probabilistic data association algorithm (CMSCJPDA) is proposed. Firstly, the multi-sensor tracking problem is decomposed into several single-sensor multi-target tracking problems, which are sequentially processed during the estimation. Then, in each sensor, the assignment of its measurements to target tracks is accomplished on the basis of joint probabilistic data association (JPDA), and a weighted probability fusion method with square root version of a cubature Kalman filter (SRCKF) is utilized to estimate the targets' state. With the measurements in all sensors processed CMSCJPDA is derived and the global estimated state is achieved. Experimental results show that CMSCJPDA is superior to the state-of-the-art algorithms in the aspects of tracking accuracy, numerical stability, and computational cost, which provides a new idea to solve multi-sensor tracking problems.

Keywords: multi-sensor tracking; data association; cubature Kalman filter; state estimation; centralized filtering

1. Introduction

The centralized state estimation method for multiple-target tracking with multiple sensors can integrate detecting information from different sensors according to certain rules, which will make the tracking results more accurate than that of a single sensor [1–3]. Multi-sensor fusion is an important data processing method in the field of target tracking, especially under nonlinear conditions, and the fused result with multiple sensors is often better for multiple-target tracking, which has been widely concerned by scholars both at home and abroad [3–12].

The state-of-the-art techniques for multiple-target tracking mainly contains two categories. The first category is the tracking method based on a random finite set; the other is the method based on measurement to track association. The former method can directly track multiple targets without associating the detected measurements with the tracks in the surveillance area. However, this kind of method faces complex integral operations, which are often difficult to find an exact solution. Although approximation methods have been a good choice for this problem, the computational cost is too great, and improper approximation will lead to degradation of the tracking accuracy, thus making it a long way from being extensively utilized in practical applications [4]. The latter method has been widely applied in various tracking systems, and is still the most common method for centralized tracking [5–14].

In the second category, the sequential multi-sensor joint probabilistic data association algorithm (MSJPDA) is a classical and effective method for multi-sensor tracking, which achieves better tracking performance compared with its parallel counterparts [10,11]. Unfortunately, current MSJPDA algorithms are often proposed to solve association problems under linear circumstances, and the nonlinear environment is seldom involved in the literature. As is known, the nonlinear equation can be linearized according to Taylor expansion, then a MSJPDA algorithm based on the extended Kalman filter (EKF), the so called MSJPDA-EKF is derived [12–14]. However, it is essential for MSJPDA-EKF to calculate the Jacobian matrix in the linearization step in each iteration, and the ignorance of higher orders may introduce large linearization errors, which may result in degraded performance, and even divergence [1,2,14]. In addition, the estimated state will affect the calculation of association probability in the next iteration, thus causing changes in the computation of weights of joint events, which may eventually make false correlations, and even lead to divergence [15]. To conquer the shortcomings of MSJPDA-EKF, the MSJPDA algorithm based on the unscented Kalman filter (MSJPDA-UKF) is proposed. MSJPDA-UKF can acquire higher estimation accuracy than MSJPDA-EKF, and is relatively more stable [16]. However, reasonable adjustments are needed in MSJPDA-UKF to achieve the ideal performance, and a numerical instability phenomenon is easy to occur in the situation of high dimensional estimation [16–18].

To address the high dimensional nonlinear estimation problem, the cubature Kalman filter (CKF) based on the third-degree spherical-radial cubature rule has been recently proposed [19–25]. In the procedure of CKF, a certain number of typical cubature points are selected to approximate the posterior probability, then the mean and covariance are captured through a nonlinear transfer function to achieve accurate estimation of the target state [19]. CKF and UKF are both moment-matching filters which deterministically select a set of weighted sample points to approximate the posterior probability density. Compared with EKF, CKF and UKF do not have to calculate the Jacobian matrix in each iteration, thereby reducing the computational cost. At the same time, they both avoid the negative influence of the truncation error in the process of nonlinear state estimation, which is particularly effective in the case of state estimation with strong nonlinearity [19,21–23]. In UKF, to achieve a better estimation performance, the adjustment of the scaling parameter is an essential step, as different scaling parameters may yield completely distinct estimation results. The selection of the scaling parameter is of great significance to the performance of UKF in some sense [16,19,21,25]. However, there is no parameter to be adjusted for CKF, and the selected sampling points and weights are only related to the dimension of the target state, which can be calculated in advance to reduce the computational complexity. Furthermore, CKF is more accurate with respect to high dimensional estimation, and easy for design and implementation [19,21–24]. Especially in the case of high dimension, EKF and UKF suffer from the curse of dimensionality or divergence, or both, but CKF still performs well [19,21,25]. Unfortunately, in practice, arithmetic operations performed on finite word-length digital computers may introduce large errors; then the symmetry and positive definiteness of the covariance are often not satisfied. Moreover, perhaps the loss of positive definiteness is more perilous as it terminates the running of the CKF. Then, the square root version of CKF, the so called SRCKF is proposed to address this problem [19,21,23,24]. In SRCKF, only the square root factors are propagated during the iterations. Therefore, the square rooting operations of the matrix are avoided, thus reducing the computation cost. The symmetry and positive definiteness of the covariance are preserved, and the numerical accuracy is also improved [19,21,25].

In this paper, to deal with the target tracking problem in a nonlinear system under a cluttered environment, a novel centralized multi-sensor square root cubature joint probabilistic data association algorithm (CMSCJPDA) is proposed. In CMSCJPDA, measurements of each single sensor are sequentially processed to compute the association probabilities based on the similar rules in JPDA, then SRCKF and the validated measurements are chosen to estimate the state of targets according to weighed state fusion. Eventually, the accurate estimation of multiple targets' states with multiple sensors is accomplished.

The rest of this paper is organized as follows: Section 2 describes the problem of tracking multiple targets with multiple sensors. Then, in Section 3, a simple overview of the numerical integral approximation method based on spherical-radial principle is introduced. The main idea of CMSCJPDA is detailed in Section 4, and an algorithm flow is given. The numerical experiments are designed in Section 5, and the comparison and analysis of CMSCJPDA against several existing methods is also presented in this section. Concluding remarks of this paper and future work are given in Section 6.

2. Problem Formulation

Consider sensors are applied to track N_t targets in a cluttered environment. For arbitrary target t ($1 \leq t \leq N_t$), $X^t(k)$ denotes the state of target t at discrete time instant k . For brevity, without the control input term, the discrete-time state equation of a nonlinear system is:

$$X^t(k + 1) = f^t[k, X^t(k)] + V^t(k) \tag{1}$$

where k is the discrete time instant, $f^t(\cdot)$ is a known nonlinear function, and $V^t(k)$ is independent zero-mean Gaussian process noise with covariance:

$$E\{V^t(k)[V^t(l)]^T\} = Q^t(k)\delta(k, l) \tag{2}$$

where $\delta(k, j)$ is the Kronecker delta function.

The measurement equation of sensor i for target t is:

$$Z^{i,t}(k) = h^i[k, X^t(k)] + W^i(k) \tag{3}$$

where $i = 1, 2, \dots, N_s$ represents the label of sensors. $Z^{i,t}(k)$ is the measurement vector of sensor i for target t at time instant k . $h^i(\cdot)$ represents a known nonlinear function. $W^i(k)$ is the zero-mean, independent of process noise in Equation (3) and independent from sensor to sensor, with covariance:

$$E\{W^i(k)[W^i(l)]^T\} = R^i(k)\delta(k, l) \tag{4}$$

3. Numerical Approximation of the Multi-Dimensional Weighted Integral

3.1. Third-Degree Spherical-Radial Rule

Before introducing the CMSCJPDA method, a brief description of the numerical integral approximation based on spherical-radial principle is given. Consider the following multi-dimensional Gaussian weighted integral:

$$I(f) = \int_{R^n} f(x) \exp(-x^T x) dx \tag{5}$$

where $f(\cdot)$ is an arbitrary function, R^n denotes the integral region. As is known, the key to addressing the problem of nonlinear filtering based on Bayesian theory is to compute the first-order and second-order moments. In other words, the core of the Bayesian filter is how to compute Gaussian weighted integral which is of the form *nonlinear function* \times *Gaussian density* that is illustrated in Equation (5).

Let $x = ry$, where $y^T y = 1$, So that $x^T x = r^2$ for $r \in [0, \infty)$. Then the integral Equation (5) can be rewritten as:

$$I(f) = \int_0^\infty \int_{U_n} f(ry) r^{n-1} \exp(-r^2) d\sigma(y) dr \tag{6}$$

where $U_n = \{y \in R^n \mid y^T y = 1\}$ is the surface of the sphere and $\sigma(\cdot)$ is the spherical surface measure or the area element on U_n . Therefore, we can split Equation (6) into two integrals:

$$I(f) = \int_0^\infty S(r)r^{n-1} \exp(-r^2)dr \tag{7}$$

$$S(r) = \int_{U_n} f(ry)d\sigma(y) \tag{8}$$

where $I(f)$ is the radial weighted integral that is shown in Equation (7), $S(r)$ is the spherical integral with the unit weighting function $\omega(y) = 1$ that is shown in Equation (8).

Then the above integrals described in Equations (7) and (8) can be solved according to the third-degree spherical cubature rule and radical rule, respectively:

$$\int_{U_n} f(y)d\sigma(y) \approx \omega \sum_{i=1}^{2n} f[u]_i \tag{9}$$

$$\int_a^b f(x)\omega(x)dx \approx \sum_{i=1}^m \omega_i f(x_i) \tag{10}$$

where $[u]_i$ represents the i -th element of generator u , and u is a generator if $u = (u_1, u_2, \dots, u_r, 0, \dots, 0) \in R^n$, where $u_i > u_{i+1} > 0, i = 1, 2, \dots, (r - 1)$. For simplicity, the $(n - r)$ zero coordinates are ignored and notation $[u_1, u_2, \dots, u_r]$ is used to represent a complete fully-symmetric set of points that can be obtained by permutating and changing the sign of generator u in all possible ways. For example, $[1] \in R^2$ denotes the following set of points:

$$[1] = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\} \tag{11}$$

where $\omega(x)$ is a known weighting function which is non-negative on the interval $[a, b]$.

As is shown in Equation (9), the spherical cubature rule indicates that the spherical integral can be approximated by a weighted sum of function values at the sample points, which are located at the intersection of the unit sphere and its axes. The radical rule described in Equation (10) indicates that an m -point Gaussian quadrature is equal to the sum of $(2m - 1)$ polynomials [21].

Especially, a standard Gaussian weighted integral can be computed based on third-degree spherical-radial rule as follows [25]:

$$I_N(f) = \int_{R^n} f(x)N(x; \mathbf{0}, \mathbf{I})dx \approx \sum_{i=1}^{2n_x} \omega_i f(\xi_i) \tag{12}$$

where:

$$\xi_i = \sqrt{\frac{2n_x}{2}} [1]_i, \omega_i = \frac{1}{2n_x}, i = 1, 2, \dots, 2n_x \tag{13}$$

where ξ_i is the cubature point, ω_i is the corresponding weight. $[1]_i$ denotes the i -th row or column of the generator $[1]$.

Actually, the weighting function term $\omega(\cdot)$ of the integrand is often not subject to the standard Gaussian distribution. In other words, the Gaussian weighted integral in nonlinear filtering cannot be

solved directly by Equation (12). Fortunately, the following equation makes it possible to work out the nonstandard Gaussian weighted integrals which occur in nonlinear condition:

$$\int_{\mathbb{R}^n} f(x)N(x; \mu, \Sigma)dx = \int_{\mathbb{R}^n} f(\sqrt{\Sigma}x + \mu)N(x; \mathbf{0}, \mathbf{I})dx \tag{14}$$

Through Equation (14), which is proved in Appendix A, the nonstandard Gaussian weighted integrals can be transformed into the standard ones. Then, the approximation of posterior mean and error covariance can be addressed by Equation (12). Thus, an iteration of the time and the measurement updates in the Bayesian filter is completed.

3.2. Accuracy Analysis

Assume an n -dimensional vector $x \sim N(\bar{x}, P)$, where \bar{x} is the mean of x , and P is the corresponding covariance. We extend the nonlinear function $f(x)$ at \bar{x} based on Taylor extension:

$$f(x) = f(\bar{x}) + D_{\Delta x}f + \frac{D_{\Delta x}^2 f}{2!} + \frac{D_{\Delta x}^3 f}{3!} + \frac{D_{\Delta x}^4 f}{4!} + \dots \tag{15}$$

where $\Delta x = x - \bar{x}$, $D_{\Delta x}f = [\Delta x]^T \nabla f(x)|_{x=\bar{x}}$, $\nabla f(x)$ is the partial derivative of $f(x)$.

- (1) We use the third-degree spherical-radial rule to approximate the multi-dimensional integral $I(f)$, then:

$$I_{CKF}(f) = \frac{1}{2n_x} \sum_{i=1}^{2n_x} f(x_i) = \frac{1}{2n_x} \sum_{i=1}^{2n_x} f(\bar{x}) + D_{\Delta x_i}f + \frac{D_{\Delta x_i}^2 f}{2!} + \frac{D_{\Delta x_i}^3 f}{3!} + \frac{D_{\Delta x_i}^4 f}{4!} + \dots \tag{16}$$

where x_i is the sampled cubature point, $\Delta x = x_i - \bar{x}$. Through proper simplification, the final result can be written as:

$$I_{CKF}(f) = f(\bar{x}) + \left(\frac{\nabla^T P \nabla}{2!} \right) f + \frac{n^{k-1}}{(2k)!} \sum_{i=1}^{n_x} \left(a_{i1} \frac{\partial}{\partial x_1} + \dots + a_{in} \frac{\partial}{\partial x_n} \right)^{2k} f(x) \Big|_{x=\bar{x}} \tag{17}$$

where $k = 1, 2, 3, \dots$, $a_{ij} = [\sqrt{P}]_{ij}$, $i, j = 1, 2, \dots, n$, $\frac{\partial}{\partial x_i} f$ is the partial derivative of $f(x)$ at the i th component of x . The interested reader can be referred to paper [26] for details in the extension.

- (2) Like the derivation process of $I_{CKF}(f)$, we use UKF to operate the approximation, and the final result is:

$$I_{UKF}(f) = f(\bar{x}) + \left(\frac{\nabla^T P \nabla}{2!} \right) f + \frac{(n + \kappa)^{k-1}}{(2k)!} \sum_{i=1}^{n_x} \left(a_{i1} \frac{\partial}{\partial x_1} + \dots + a_{in} \frac{\partial}{\partial x_n} \right)^{2k} f(x) \Big|_{x=\bar{x}} \tag{18}$$

where κ is the scaling parameter, and the remaining notations are the same as Equation (17). To capture the kurtosis of the prior density as correctly as possible, κ is suggested to be $\kappa = 3 - n_x$ [16].

This paper is to tackle the high-dimensional estimation problem, so here we just discuss the case in which the state dimension $n_x > 3$. As is shown in Equations (17) and (18), the first two terms in both equations are the same, the only difference is in the last term. If $n_x > 3$, $n^{k-1} > (n + \kappa)^{k-1}$, and $2k$ is an even number, then $I_{CKF}(f) > I_{UKF}(f)$, which indicates that CKF is more accurate than UKF in high-dimensional estimation.

We choose the stability factor $I = \sum_i |\omega_i| / \sum_i \omega_i$ defined as the measure of the numerical stability of the multi-dimensional integral, where ω_i is the sampling weight. It is proven that the sampling rule implemented in a finite-precision arithmetic machine introduces a large amount of round-off errors when the stability factor I is larger than unity [19].

In UKF, if $n_x > 3$, and $\kappa + n_x = 3$, so $\kappa = 3 - n_x < 0$, then $\omega_0 = 1 - n_x/3 < 0$. The stability factor is $I_{UKF} = \frac{2n_x}{3} - 1 > 1$, and it scales linearly with state dimension n_x , which indicates that UKF introduces significant perturbations in numerical estimates for the moment integral. However, in CKF, the stability factor I_{CKF} always meets $I_{CKF} = 1$, which shows that CKF is more accurate than UKF for high-dimensional state estimation.

4. Centralized Multi-Sensor Square Root Cubature Joint Probabilistic Data Association

In CMSCJPDA, measurements of each sensor are processed in sequence, then the multi-sensor multi-target tracking problem can be reduced to several single-sensor multi-target tracking problems, which are easier to solve.

Assume there are $m_{i,k}^t$ validated measurements for target t obtained by sensor i at time instant k . l_i ($0 \leq l_i \leq m_{i,k}^t$) is the label of validated measurements from sensor i , $l_i = j$ ($j \neq 0$) represents the j th measurements in the validated region. Especially, $l_i = 0$ indicates that there is no measurement in the validated region. $Z_{l_i}^{i,t}(k)$ represents the l_i th validated measurement, $\hat{Z}_{|l_i-1}^t(k|k)$ represents the predicted measurement, which will be discussed in detail later. $\beta_{i,l_i}^t(k)$ represents the association probability that the measurement l_i originated from target t , $K_i^t(k)$ is the filtering gain. The state estimate and the corresponding error covariance after processing the measurements of the i th sensor are denoted by $\hat{X}_i^t(k|k)$ and $P_i^t(k|k)$, respectively. $\hat{X}_0^t(k|k)$ and $P_0^t(k|k)$ represent the initial estimation, and $\hat{X}^t(k|k)$ and $P^t(k|k)$ represent the final estimation at time instant k , respectively. Then, the state update is as follows:

$$\hat{X}_i^t(k|k) = \hat{X}_{i-1}^t(k|k) + K_i^t(k) \sum_{l_i=0}^{m_{i,k}^t} \beta_{i,l_i}^t(k) [Z_{l_i}^{i,t}(k) - \hat{Z}_{|l_i-1}^t(k|k)] \tag{19}$$

where:

$$\begin{cases} \hat{X}_0^t(k|k) = \hat{X}^t(k|k-1), \hat{X}^t(k|k) = \hat{X}_{N_s}^t(k|k) \\ P_0^t(k|k) = P^t(k|k-1), P^t(k|k) = P_{N_s}^t(k|k) \end{cases} \tag{20}$$

Then, the update of the error covariance is:

$$\begin{aligned} P_i^t(k|k) = & P_{i-1}^t(k|k) - [1 - \beta_{0,i}^t(k)] K_i^t(k) S_i^t(k) [K_i^t(k)]^T \\ & + \sum_{l_i=0}^{m_{i,k}^t} \beta_{i,l_i}^t(k) \hat{X}_{i,l_i}^t(k|k) [\hat{X}_{i,l_i}^t(k|k)]^T - \hat{X}_i^t(k|k) [\hat{X}_i^t(k|k)]^T \end{aligned} \tag{21}$$

Note that the key point for nonlinear single sensor multi-target tracking problem is to compute the association probability $\beta_{i,l_i}^t(k)$ and nonlinear state estimation $\hat{X}_i^t(k|k)$. JPDA is thought to be an effective method for multiple targets tracking with single sensor, so we choose JPDA to compute $\beta_{i,l_i}^t(k)$ here. The remaining problem is to obtain $\hat{X}_i^t(k|k)$. Although CKF is a good method to deal with the nonlinear state estimation problem, it is easily influenced by limited computer word-length and numerical errors, which may lead to loss of positive definiteness and symmetry of the error covariance matrix [19,21–24]. The effective way to preserve both properties and improve the numerical stability is to design a square root version of the CKF. Despite the fact that the square root cubature Kalman filter (SRCKF) is reformulated to propagate the square roots of the covariance matrices, both CKF and SRCKF are algebraically equivalent to each other, the interested reader is referred to [19]. In the proposed method, we choose SRCKF to update the state of the targets after the correlation step.

4.1. Computation of Association Probabilities

The probability $\beta_{l_i, l_i}^t(k)$ that measurement l_i originated from target t can be calculated by summing over all feasible events for which it is true [1]:

$$\beta_{l_i, l_i}^t(k) = \sum_{m=1}^{n_i^t(k)} \hat{\omega}_{l_i, t}^m(\theta_m(k)) \Pr\{\theta_m(k) | Z_i^t(k)\} \tag{22}$$

where $n_i^t(k)$ is the number of joint events, $\theta_m(k)$ denotes the event m , and $\hat{\omega}_{l_i, t}^m(\theta_m(k))$ is the following binary element.

$$\hat{\omega}_{l_i, t}^m(\theta_m(k)) = \begin{cases} 1, & \text{if } \theta_m(k) \text{ occurs} \\ 0, & \text{else} \end{cases} \tag{23}$$

$\Pr\{\theta_m(k) | Z_i^t(k)\}$ represents the posterior probability of event m , which can be computed by Equation (24):

$$\Pr\{\theta_m(k) | Z_i^t(k)\} = \frac{1}{C} \frac{\phi[\theta_m(k)]!}{V^{\phi[\theta_m(k)]}} \cdot \prod_{l_i=1}^{m_{i,k}^t} N_{l_i}^{i,t} [Z_{l_i}^{i,t}(k)]^{\tau_{l_i}[\theta_m(k)]} \prod_{t=1}^{N_t} (P_D^t)^{\delta_t[\theta_m(k)]} (1 - P_D^t)^{1 - \delta_t[\theta_m(k)]} \tag{24}$$

where C is a constant, $\phi[\theta_m(k)]$ is the total number of false measurements in event m . V is the volume of the entire surveillance region, and P_D^t is the detection probability of target t . The target detection indicator $\delta_t[\theta_m(k)]$ is defined as:

$$\delta_t[\theta_m(k)] = \sum_{l_i=1}^{m_{i,k}^t} \hat{\omega}_{l_i, t}^m(\theta_m(k)) = \begin{cases} 1, & \text{if target } t \text{ is detected} \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

which indicates whether any measurement is correlated with target t in event m . In the same way, it is also easy to define measurement association indicator:

$$\tau_{l_i}[\theta_m(k)] = \sum_{t=1}^{N_t} \hat{\omega}_{l_i, t}^m(\theta_m(k)) = \begin{cases} 1, & \text{if measurement } l_i \text{ is associated with a target} \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

which indicates whether measurement l_i is correlated with a target in event m . $N_{l_i}^{i,t} [Z_{l_i}^{i,t}(k)]$ is a conditional probability density function under Gaussian assumption, i.e.:

$$N_{l_i}^{i,t} [Z_{l_i}^{i,t}(k)] = \frac{1}{\sqrt{|2\pi S_i^t(k)|}} \exp\left\{-\frac{1}{2} [Z_{l_i}^{i,t}(k) - \hat{Z}_{l_i|l_i-1}^t(k|k)]^T \cdot [S_i^t(k)]^{-1} [Z_{l_i}^{i,t}(k) - \hat{Z}_{l_i|l_i-1}^t(k|k)]\right\} \tag{27}$$

4.2. State Update

During the update of state, the data processing in first sensor is a little different from that in other sensors. Firstly, the specific procedures for the state update in sensors with label $i > 1$ are as follows:

Step 1: Calculate the cubature points ($j = 1, 2, \dots, 2n_x$):

$$\hat{X}_{l_i|l_i-1}^t(k|k) = \hat{X}_{l_i-1|l_i-1}^t(k|k) \tag{28}$$

$$X_{j, l_i|l_i-1}^t(k|k) = S_{l_i|l_i-1}^t(k|k) \xi_j + \hat{X}_{l_i|l_i-1}^t(k|k) \tag{29}$$

where n_x is the dimension of the state. $S_{i|i-1}^t(k|k)$ is the square root factor of $P_{i-1}^t(k|k)$, i.e.:

$$P_{i-1}^t(k|k) = S_{i|i-1}^t(k|k) \left[S_{i|i-1}^t(k|k) \right]^T \tag{30}$$

Equations (28) and (29) indicate that the estimated state $\hat{X}_{i-1|i-1}^t(k|k)$ and error covariance $P_{i-1}^t(k|k)$ of the previous sensor are regarded as the predicted state $\hat{X}_{i|i-1}^t(k|k)$ and error covariance $P_{i|i-1}^t(k|k)$ of the next sensor, which is the greatest difference of the sensors with label $i > 1$ compared with the first sensor with label $i = 1$ in the process of state estimation. That is to say, the time update is not needed in sensors with label $i > 1$, and the estimation of the previous sensor is used as the prediction of the next sensor.

Step 2: Evaluate the prediction of cubature points:

$$Z_{j,i|i-1}^t(k|k) = h^i \left[k, X_{j,i|i-1}^t(k|k) \right] \tag{31}$$

Step 3: Estimate the prediction of the corresponding measurement:

$$\hat{Z}_{i|i-1}^t(k|k) = \frac{1}{2n_x} \sum_{j=1}^{2n_x} Z_{j,i|i-1}^t(k|k) \tag{32}$$

Step 4: Estimate the square-root of the innovation covariance:

$$S_{zz, i|i-1}^t(k|k) = \text{Tria} \left(\left[Z_{i|i-1}^t(k|k) \quad S_{R^i(k)} \right] \right) \tag{33}$$

where $S_{zz, i|i-1}^t(k|k)$ is a lower triangular matrix. $S_{R^i(k)}$ denotes a square root factor of $R^i(k)$ that $R^i(k) = S_{R^i(k)} \left[S_{R^i(k)} \right]^T$, and the weighed, centred matrix:

$$\mathcal{Z}_{i|i-1}^t(k|k) = \frac{1}{\sqrt{2n_x}} \left[Z_{1,i|i-1}^t(k|k) - \hat{Z}_{i|i-1}^t(k|k), Z_{2,i|i-1}^t(k|k) - \hat{Z}_{i|i-1}^t(k|k), \dots, Z_{2n_x,i|i-1}^t(k|k) - \hat{Z}_{i|i-1}^t(k|k) \right] \tag{34}$$

Note that $S = \text{Tria}(A)$ denotes the QR decomposition, where S is a lower triangular matrix. The relationship of matrices S and A is as follows: Let B be an upper triangular matrix obtained from the QR decomposition on A^T . Then, the lower triangular matrix S is obtained as $S = R^T$.

Step 5: Estimate the cross-covariance matrix:

$$P_{xz,i|i-1}^t(k|k) = \mathcal{X}_{i|i-1}^t(k|k) \left[\mathcal{Z}_{i|i-1}^t(k|k) \right]^T \tag{35}$$

where:

$$\mathcal{X}_{i|i-1}^t(k|k) = \frac{1}{\sqrt{2n_x}} \left[X_{1,i|i-1}^t(k|k) - \hat{X}_{i|i-1}^t(k|k), X_{2,i|i-1}^t(k|k) - \hat{X}_{i|i-1}^t(k|k), \dots, X_{2n_x,i|i-1}^t(k|k) - \hat{X}_{i|i-1}^t(k|k) \right] \tag{36}$$

Step 6: Estimate the filter gain:

$$K_i^t(k) = P_{xz,i|i-1}^t(k|k) / S_{zz, i|i-1}^t(k|k) / \left[S_{zz, i|i-1}^t(k|k) \right]^T \tag{37}$$

Step 7: Update the nonlinear state:

$$\hat{X}_{i,i}^t(k|k) = \hat{X}_{i-1}^t(k|k) + K_i^t(k) \left[Z_{i,i}^t(k) - \hat{Z}_{i|i-1}^t(k|k) \right] \tag{38}$$

Equations (28)–(38) give a solution to the estimation of the target state and error covariance at time instant k with sensor label $i > 1$.

Now the estimation problem in the first sensor with label $i = 1$ is considered, the process of the measurement update is the same as that in sensors with label $i > 1$, but the time update is quite different. In the sensor with label $i = 1$, the estimation of prediction of the state and error covariance is essential. The current cubature points are:

$$X_j^t(k-1|k-1) = S^t(k-1|k-1)\xi_j + \hat{X}^t(k-1|k-1) \tag{39}$$

The propagated cubature points are:

$$X_j^t(k|k-1) = f^t \left[k-1, X_j^t(k-1|k-1) \right] \tag{40}$$

Then the predicted state is:

$$\hat{X}^t(k|k-1) = \frac{1}{2n_x} \sum_{j=1}^{2n_x} X_j^t(k|k-1) \tag{41}$$

and the predicted error covariance is:

$$P^t(k|k-1) = \frac{1}{2n_x} \sum_{j=1}^{2n_x} \left[X_j^t(k|k-1) - \hat{X}^t(k|k-1) \right] \cdot \left[X_j^t(k|k-1) - \hat{X}^t(k|k-1) \right]^T + Q^t(k) \tag{42}$$

Equations (39)–(42) describe the time update in the first sensor with label $i = 1$. With the measurement update process composed of Equations (28)–(38) applied, the estimated state $\hat{X}_1^t(k|k)$ and error covariance $P_1^t(k|k)$ can be achieved respectively after the data of the first sensor is processed.

To state the proposed method more clearly, the algorithm flow of CMSCJPDA is illustrated in Table 1.

Table 1. Algorithm flow of CMSCJPDA.

Input: The estimated state $\hat{X}^t(k-1 k-1)$ and error covariance $P^t(k-1 k-1)$ at time instant $k-1$;
Output: The estimated state $\hat{X}^t(k k)$ and error covariance $P^t(k k)$ at time instant k ;
Step 1. Time update ($i = 1$)
Compute the predicted state $\hat{X}^t(k k-1)$ and error covariance $P^t(k k-1)$ according to Equations (39)–(42);
Step 2. Initialization
$\hat{X}_0^t(k k) = \hat{X}^t(k k-1)$, $P_0^t(k k) = P^t(k k-1)$;
Step 3. Measurement update
For $t = 1 : N_t$
For $i = 1 : N_s$
Compute the estimation of state $\hat{X}_i^t(k k)$ and error covariance $P_i^t(k k)$ with the validated measurements of sensor i according to Equations (19)–(38);
End
End
Step 4. Establish the final estimation
$\hat{X}^t(k k) = \hat{X}_{N_s}^t(k k)$, $P^t(k k) = P_{N_s}^t(k k-1)$

5. Numerical Experiments

In this section, the numerical results and the effectiveness of the proposed CMSCJPDA algorithm are discussed in two simulation scenarios: crossing target tracking and maneuvering target tracking. The performance of CMSCJPDA is compared with MSJPDA-EKF and MSJPDA-UKF in two scenarios.

For a fair comparison, we make 50 independent Monte Carlo runs. In each run, the measurement noises are generated independently, and the corresponding noisy position measurements are used to initialize the estimate, which guarantees consistency of the initialization of the filter and makes

the final estimate unbiased. Except for the initialization, the process of each Monte Carlo run is the same, and the mean value of 50 runs is considered as the final result. The total number of steps per run is 100. $T = 1$ s is the sampling interval. The nonparametric model is used for the probability mass function (PMF) of the number of false measurements. The expected number of false measurements in the validation gate is $m = 2$. The detection probabilities of all targets are assumed to be $P_D = 0.9$. The probability mass is assumed to be $P_G = 0.9997$. Due to the large initial error, we display the numerical results after the tenth step for clarity.

Targets in the surveillance area are observed by three two-dimensional sensors. The measurement equation of sensor i is:

$$Z^i(k) = \begin{bmatrix} \sqrt{(x(k) - x_{pi})^2 + (y(k) - y_{pi})^2} \\ \arctan\left(\frac{y(k) - y_{pi}}{x(k) - x_{pi}}\right) \end{bmatrix} + W^i(k) \tag{43}$$

where $i = 1, 2, 3$ is the sensor label. $(x(k), y(k))$ is the true state of the targets, (x_{pi}, y_{pi}) is the position of sensor i . Other parameters of the sensors are set as shown in Table 2.

Table 2. Sensor position and parameter setting.

Sensor Labe	Sensor Position (m)	Ranging Error (m)	Angle Error (rad)
1	(0, 0)	100	0.01
2	(-500, -500)	200	0.02
3	(-500, 500)	300	0.03

Performance metrics: To compare the tracking performance of different algorithms, the root mean square error (RMSE) is chosen as the metric. The root mean square position error of target t at time instant k is defined as:

$$RMSE_{pos}(k, t) = \sqrt{\frac{1}{M} \sum_{i=1}^M (x_i^t(k) - \hat{x}_i^t(k|k))^2} \tag{44}$$

where M is the total number of Monte Carlo simulations. $x_i^t(k)$ and $\hat{x}_i^t(k|k)$ are true and estimated positions of target t in the x direction in the n -th Monte Carlo run. Similarly, we may also define formulas of the root mean square velocity error.

5.1. Crossing Targets Tracking

5.1.1. Simulation Scenario

Consider a two-crossing target tracking case. The state model is:

$$X(k+1) = F(k)X(k) + \Gamma(k)V(k) \tag{45}$$

where the component of process noise $q_1 = q_2 = 0.01$. The state transition matrix $F(k)$ is defined as:

$$F(k) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{46}$$

and the process noise distribution matrix is:

$$\Gamma(k) = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \tag{47}$$

The initial states of two targets are $X_1(0) = [-29,500 \text{ m}, 400 \text{ m/s}, 34,500 \text{ m}, 2,212,400 \text{ m/s}]^T$, $X_2(0) = [-26,500 \text{ m}, 296 \text{ m/s}, 34,500 \text{ m}, -400 \text{ m/s}]^T$. After about 31 s, the two targets cross at the point $(-17,000 \text{ m}, 22,000 \text{ m})$.

5.1.2. Results and Analysis

The true and filtered tracks of two crossing targets are illustrated in Figure 1, which shows the tracking performance of three algorithms in a single run.

To give a more explicit picture of the estimated results, the root mean square position error of three data association algorithms in the x and y directions is shown in Figures 2 and 3, while the root mean square velocity error in the x and y directions is shown in Figures 4 and 5. As is shown, the three algorithms can effectively estimate the state of both targets. However, CMSCJPDA and MSJPDA-UKF, which are based on deterministic sampling, are of higher tracking accuracy than MSJPDA-EKF, and CMSCJPDA is slightly more accurate than MSJPDA-UKF, which just validates the argument in [19]. With the square root version of CKF applied in the proposed method, the square rooting operations of covariance matrix is avoided and the tracking performance is effectively enhanced.

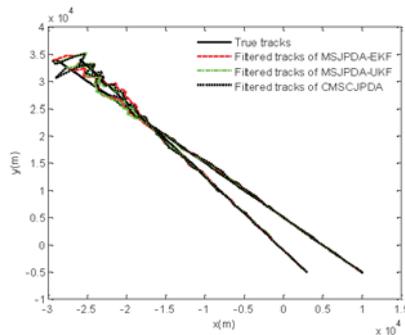


Figure 1. True tracks and filtered tracks of two crossing targets.

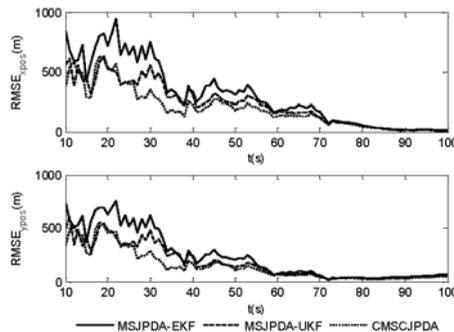


Figure 2. Root mean square position error of target 1.

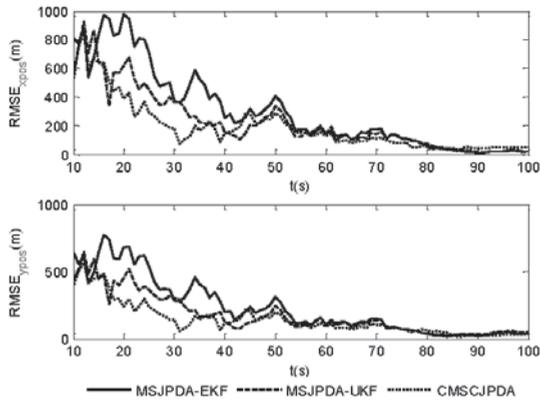


Figure 3. Root mean square position error of target 2.

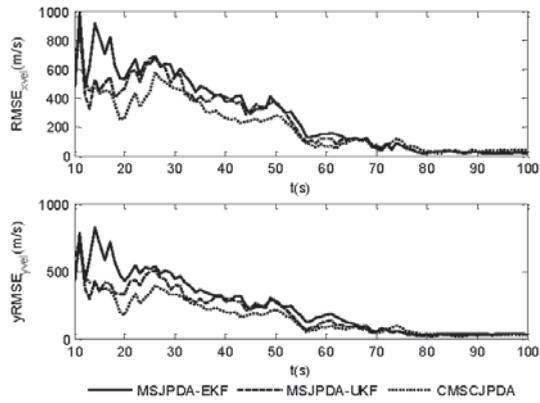


Figure 4. Root mean square velocity error of target 1.

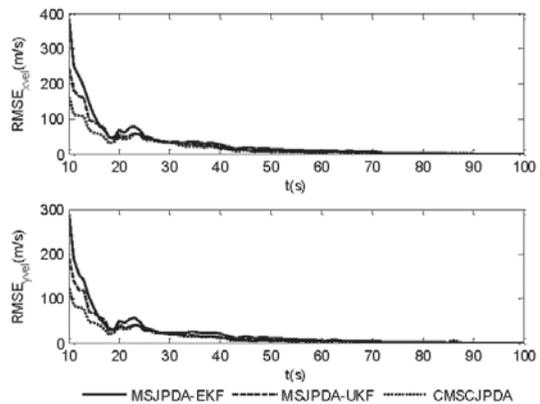


Figure 5. Root mean square velocity error of target 2.

To further validate the effectiveness of the proposed algorithm, numerical stability and computational complexity of the three algorithms are compared in Table 3, which gives the average

divergence times (ADT) and average time consumption (ATC) in 50 Monte Carlo runs. As is shown, the ADT of MSJPDA-EKF is twice more than that of MSJPDA-UKF, and nearly four times more than that of CMSCJPDA, to some extent CMSCJPDA is relatively more stable in terms of convergence. As for the computational cost, the ATC of CMSCJPDA is almost half that of MSJPDA-EKF, and less than that of MSJPDA-UKF. In a word, compared with MSJPDA-EKF and MSJPDA-UKF, CMSCJPDA can achieve relatively higher tracking accuracy, and is more stable and less computationally complex under a nonlinear cluttered environment.

Table 3. Performance comparison of the three algorithms.

Algorithms	Average Divergence Times	Average Time Consumption (s)
MSJPDA-EKF	0.87	0.423
MSJPDA-UKF	0.42	0.346
CMSCJPDA	0.27	0.257

5.2. Maneuvering Targets Tracking

5.2.1. Simulation Scenario

Assume there are two maneuvering targets in the surveillance area. The initial states of two targets are $X_1(0) = [20,000 \text{ m}, -600 \text{ m/s}, 1800 \text{ m}, 500 \text{ m/s}]^T$, $X_2(0) = [4000 \text{ m}, 600 \text{ m/s}, 1800 \text{ m}, 200 \text{ m/s}]^T$. The two targets move at a constant speed in a straight line for 30 s. Then, from 30 s to 50 s, they both move with constant accelerations of $a_{1x} = -30 \text{ m/s}^2$, $a_{1y} = 20 \text{ m/s}^2$, $a_{2x} = 30 \text{ m/s}^2$, $a_{2y} = 20 \text{ m/s}^2$ in the x and y directions, respectively. From 50 s to 70 s, they still move with constant accelerations, and the accelerations for both targets in the x and y directions are $a_{x1} = -30 \text{ m/s}^2$, $a_{y1} = -20 \text{ m/s}^2$, $a_{x2} = 30 \text{ m/s}^2$, $a_{y2} = -20 \text{ m/s}^2$, respectively. After 70 s, they both move at a constant speed again until 100 s. The true trajectories of both targets are illustrated in Figure 6.

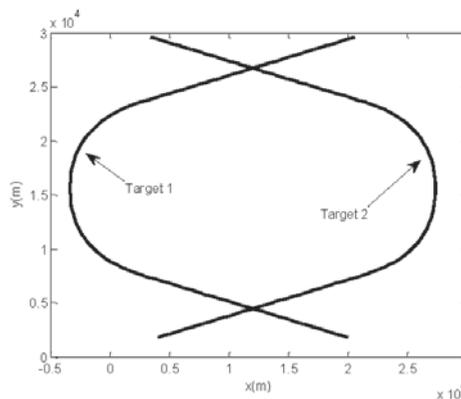


Figure 6. The trajectories of two cross-maneuvering targets.

5.2.2. Results and Analysis

The root mean square (RMS) position error of two targets is compared in Figures 7 and 8. The results show that the estimation error of CMSCJPDA and MSJPDA-UKF is much smaller than that of MSJPDA-EKF in the whole simulation. When t is about 40 s, the estimation error of MSJPDA-EKF is about 200 m larger compared with CMSCJPDA and MSJPDA-UKF. Moreover, with the new association strategy for maneuvering target tracking, CMSCJPDA has relatively higher accuracy than MSJPDA-UKF. At some time instants, the RMS position error of CMSCJPDA is about 50 m smaller than that of MSJPDA-UKF, and it converges relatively faster.

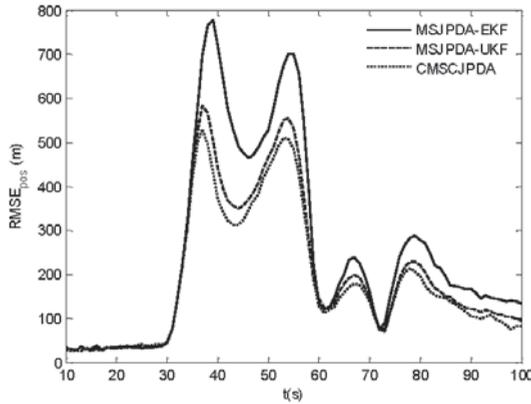


Figure 7. Root mean square position error of target 1.

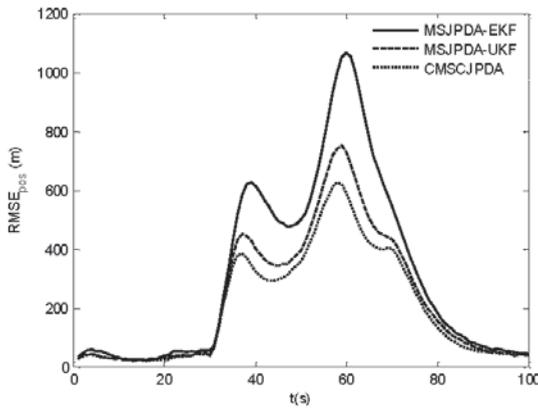


Figure 8. Root mean square position error of target 2.

Table 4 evaluates the tracking performance of the three algorithms in the aspects of mean RMS position error (MRMSE) and correct association rate (CAR). The results indicate that both CMSCJPDA and MSJPDA-UKF are more accurate than MSJPDA-EKF, and CMSCJPDA is slightly more accurate than MSJPDA-UKF. Meanwhile, the CAR of CMSCJPDA is the highest among the three algorithms, which is 11.1% higher than MSJPDA-UKF and 28.7% higher than MSJPDA-EKF, and CMSCJPDA is more time-saving.

Table 4. Performance comparison of three algorithms.

Algorithms	MRMSE (m)	CAR (%)	TC (s)
MSJPDA-EKF	206.2	47.6	0.735
MSJPDA-UKF	132.6	65.2	0.563
CMSCJPDA	104.3	76.3	0.432

6. Conclusions

In this paper, we propose a centralized multi-sensor square root cubature joint probabilistic data association algorithm, which is based on the spherical-radial principle and the sequential update scheme. Compared with the state-of-the-art method in two tracking scenarios, the proposed method is much more accurate than MSJPDA-EKF, and a slightly better than MSJPDA-UKF by using the square

root version of cubature Kalman filter to estimate the target state. Furthermore, the experimental results also show that the proposed method is less prone to divergence than MSJPDA-EKF and MSJPDA-UKF, and is more stable with respect to its numerical characteristics. Although the proposed method is less time consuming among the compared methods, due to complicated calculation of the association probability, the joint probabilistic data association-based methods are of significant computational cost, which is still less practical in real applications. Currently, the sensors in our experiment are assumed to be synchronized sampling, and there is no time difference in measurements from different sensors. However, the actual applications can rarely meet this condition, and in the future we will pay more attention to reducing the computational cost and improving the real-time performance. At the same time, the tracking problem with asynchronous sampling and changing the number of targets also deserves consideration.

Acknowledgments: The authors give their sincere thanks to the editors and the anonymous reviewers for their constructive comments of the manuscript. This study was financially co-supported by the State Key Program for Basic Research of China (no. 613XXXXX), the National Natural Science Foundation of China (nos. 61471383, 91538201, 61531020, and 61671463), the Young Elite Scientist Sponsorship Program by CAST (YESS) (no. 2015QNRC001), and China Postdoctoral Science Foundation (2017M610032).

Author Contributions: All the listed authors contributed to the research design; Y.L., J.L., G.L. and L.Q. collected the data; Y.L., J.L. and G.L. analyzed the data; Y.L. and J.L. contributed to the formula derivation; Y.L. and G.L. designed the simulation experiments; J.L. and L.Q. designed the figures and tables; Y.L. searched the literature; Y.L., J.L. and G.L. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Equation (14)

For simplicity, define the weighting function $\omega_1(x) = N(x; \mathbf{0}, \mathbf{I})$ and $\omega_2(x) = N(x; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Consider the left side of Equation (14). Since $\boldsymbol{\Sigma}$ is a positive definite matrix, then $\boldsymbol{\Sigma}$ can be factorized to be $\boldsymbol{\Sigma} = \sqrt{\boldsymbol{\Sigma}}(\sqrt{\boldsymbol{\Sigma}})^T$. Making a variable substitution by $x = \sqrt{\boldsymbol{\Sigma}}\mathbf{y} + \boldsymbol{\mu}$, we have:

$$\begin{aligned} \int_{\mathbf{R}^n} f(x)\omega_2(x)dx &= \int_{\mathbf{R}^n} f(\sqrt{\boldsymbol{\Sigma}}\mathbf{y} + \boldsymbol{\mu}) \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left\{-\left(\sqrt{\boldsymbol{\Sigma}}\mathbf{y}\right)^T \boldsymbol{\Sigma}^{-1} \left(\sqrt{\boldsymbol{\Sigma}}\mathbf{y}\right)\right\} |\sqrt{\boldsymbol{\Sigma}}| d\mathbf{y} \\ &= \int_{\mathbf{R}^n} f(\sqrt{\boldsymbol{\Sigma}}\mathbf{y} + \boldsymbol{\mu}) \frac{1}{\sqrt{|2\pi\mathbf{I}|}} \exp(-\mathbf{y}^T \mathbf{y}) d\mathbf{y} \\ &= \int_{\mathbf{R}^n} f(\sqrt{\boldsymbol{\Sigma}}\mathbf{y} + \boldsymbol{\mu}) N(\mathbf{y}; \mathbf{0}, \mathbf{I}) d\mathbf{y} \\ &= \int_{\mathbf{R}^n} f(\sqrt{\boldsymbol{\Sigma}}x + \boldsymbol{\mu}) \omega_1(x) dx \end{aligned} \quad (A1)$$

Substituting $\omega_1(x)$ and $\omega_2(x)$ into Equation (A1), we have:

$$\int_{\mathbf{R}^n} f(x)N(x; \boldsymbol{\mu}, \boldsymbol{\Sigma})dx = \int_{\mathbf{R}^n} f(\sqrt{\boldsymbol{\Sigma}}x + \boldsymbol{\mu})N(x; \mathbf{0}, \mathbf{I})dx \quad (A2)$$

References

1. Bar-Shalom, Y. *Multitarget-Multisensor Tracking: Advanced Application*; Artech House: Norwood, MA, USA, 1990.
2. He, Y.; Wang, G.; Lu, D.; Peng, Y. *Multisensor Information Fusion with Applications*, 2nd ed.; Publishing House of Electronics Industry: Beijing, China, 2010.
3. Zhang, K.; Shan, G.; Ji, B.; Chen, H. Maneuvering targets tracking based on pose-aided nonlinear filter algorithm. *Acta Electron. Sin.* **2012**, *40*, 1670–1675.
4. Liu, J.; Liu, Y.; He, Y.; Sun, S. Multi-sensor fuzzy data association based on azimuth and velocity. *Syst. Eng. Electron.* **2016**, *38*, 2040–2047.
5. Ge, Q.; Wei, Z.; Cheng, T.; Chen, C.; Wang, X. Flexible fusion structure-based performance optimization learning for multisensory target tracking. *Sensors* **2017**, *17*, 1045. [CrossRef] [PubMed]

6. Liu, Z.; Pan, Q.; Jean, D.; Gregoire, M. Hybrid Classification System for Uncertain Data. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *10*, 2783–2790. [CrossRef]
7. Taghavi, E.; Tharmarasa, R.; Kirubarajan, T.; Bar-Shalom, Y.; McDonald, M. A practical bias estimation algorithm for multisensor-multitarget tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 2–19. [CrossRef]
8. Fantacci, C.; Papi, F. Scalable multisensor multitarget tracking using the marginalized-glm density. *IEEE Signal Process. Lett.* **2016**, *23*, 863–867. [CrossRef]
9. Si, W.; Wang, L.; Qu, Z. A measurement-driven adaptive probability hypothesis density filter for multitarget tracking. *Chin. J. Aeronaut.* **2015**, *28*, 1689–1698. [CrossRef]
10. O'Neil, S.D.; Pao, L.Y. Multisensor fusion algorithm for tracking. In Proceedings of the American Control Conference, San Francisco, CA, USA, 2–4 June 1993; pp. 859–863.
11. Pao, L.Y.; Frei, C.W. A comparison of parallel and sequential implementation of a multisensory tracking algorithm. In Proceedings of the American control Conference, Seattle, WA, USA, 21–23 June 1995; pp. 1683–1687.
12. Bar-shalom, Y. *Multitarget-Multisensor Tracking: Principles and Techniques*; YBS Publishing: Stors, CT, USA, 1995.
13. Zhang, J.; He, Y.; Xiong, W. Centralized interacted multiple model multisensory fuzzy joint probabilistic data association algorithm. *Acta Electron. Sin.* **2008**, *36*, 1655–1659.
14. Han, C.; Zhu, Y.; Duan, Z.; Han, D.; Liu, W.; Yu, X. *Multi-Source Information Fusion*, 2nd ed.; Tsinghua University Press: Beijing, China, 2010.
15. Guan, X.; Zhou, X.; Rui, G. Centralized multisensor unscented joint probabilistic data association algorithm. *Syst. Eng. Electron.* **2009**, *31*, 2602–2606.
16. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [CrossRef]
17. Gao, X.; Chen, J.; Tao, D.; Li, X. Multi-sensor centralized fusion without measurement noise covariance by variational Bayesian approximation. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 718–727. [CrossRef]
18. Wang, X.; Pan, Q. Overview of deterministic sampling filtering algorithms for nonlinear system. *Control Decis.* **2012**, *27*, 801–812.
19. Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [CrossRef]
20. Arasaratnam, I.; Haykin, S. Cubature Kalman smoothers. *Automatica* **2011**, *47*, 2245–2450. [CrossRef]
21. Liu, Y.; He, Y.; Wang, H. Squared-root cubature information consensus filter for non-linear decentralised state estimation in sensor networks. *IET Radar Sonar Navig.* **2014**, *8*, 931–938.
22. Leong, P.H.; Arulampalam, S.; Lamahewa, T.A.; Abhayapala, T.D. A Gaussian-sum based cubature Kalman filter for bearings-only tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 1161–1174. [CrossRef]
23. Liu, Y.; Dong, K.; Wang, H.; Liu, J.; He, Y.; Pan, L. Adaptive Gaussian sum squared-root cubature Kalman filter with split-merge scheme for state estimation. *Chin. J. Aeronaut.* **2014**, *27*, 1242–1250. [CrossRef]
24. Liu, G.; Florentin, W.; Irene, M. Square-root sigma-point information filtering. *IEEE Trans. Autom. Control* **2012**, *57*, 2945–2950. [CrossRef]
25. Jia, B.; Sun, T.; Xin, M. Iterative diffusion-based distributed cubature Gaussian mixture filter for multisensory estimation. *Sensors* **2016**, *16*, 1741. [CrossRef] [PubMed]
26. Sun, F.; Tang, L. Estimation precision comparison of cubature Kalman filter and unscented Kalman filter. *Control Decis.* **2013**, *28*, 303–308.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Integrated Display and Simulation for Automatic Dependent Surveillance–Broadcast and Traffic Collision Avoidance System Data Fusion

Yanran Wang, Gang Xiao * and Zhouyun Dai

School of Aeronautics and Astronautics, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang District, Shanghai 200240, China; yrwang501@163.com (Y.W.); m18792833886@163.com (Z.D.)

* Correspondence: xiaogang@sjtu.edu.cn; Tel.: +86-21-3420-6192

Received: 31 August 2017; Accepted: 8 November 2017; Published: 13 November 2017

Abstract: Automatic Dependent Surveillance–Broadcast (ADS-B) is the direction of airspace surveillance development. Research analyzing the benefits of Traffic Collision Avoidance System (TCAS) and ADS-B data fusion is almost absent. The paper proposes an ADS-B minimum system from ADS-B In and ADS-B Out. In ADS-B In, a fusion model with a variable sampling Variational Bayesian-Interacting Multiple Model (VSVB-IMM) algorithm is proposed for integrated display and an airspace traffic situation display is developed by using ADS-B information. ADS-B Out includes ADS-B Out transmission based on a simulator platform and an Unmanned Aerial Vehicle (UAV) platform. This paper describes the overall implementation of ADS-B minimum system, including theoretical model design, experimental simulation verification, engineering implementation, results analysis, etc. Simulation and implementation results show that the fused system has better performance than each independent subsystem and it can work well in engineering applications.

Keywords: ADS-B; TCAS; integrated display; data fusion; airspace surveillance

1. Introduction

ADS-B [1,2] is defined by the U.S. Federal Aviation Administration (FAA) as the foundation of free flight. It is an applicable and accurate airspace surveillance technology. It can be used in traffic collision avoidance, surveillance, auxiliary approach and may have a great effect on all those aspects. TCAS [3,4] is an airborne traffic alarm and collision avoidance system which is independent of air traffic control on the ground.

The fusion of ADS-B and TCAS data can improve the prediction accuracy of TCAS systems [5], increase the rate of true alarms and decrease the rate of false and missed alarms. There are many benefits for airspace alarm accuracy and flight safety by combining with TCAS II and ADS-B system. Even if essential information of a single system is missing, the system will work and not be affected too much. The application of these technologies is closely related to the next generation of air traffic systems, which are expected to have many benefits for the manufacture and use of civil aircraft.

Regarding the ADS-B system aspects, Kunzi [3] analyzed the benefits of ADS-B in general aviation and pointed out that ADS-B will be the basis of the future surveillance system. Wang [6] designed and implemented a complete mini ADS-B air traffic control system based on five aspects which are monitoring data displaying, flight data displaying, ADS-B message processing, alarm computing and equipment running status monitoring. McCallie [7] at the Air Force Institute of Technology performed a safety analysis of the next generation of air traffic transportation. Mueller [8] proposed Aircraft ADS-B Intent Verification based on a Kalman Tracking Filter in order to improve the accuracy of ADS-B data transmission and to reduce the data error rate. Peng [9] proposed the CPR algorithm for 1090ES ADS-B system encoding and decoding to improve transmission efficiency. Purton [10] solved some of

the uncertainties in ADS-B. The procedural remedies are proposed for technical problems. Purton also focused on technical solutions for the vulnerabilities and threats in the transmission and computation.

In the aspects of ADS-B and TCAS fusion, Ni [11] studied a ADS-B and TCAS II system fusion algorithm. The Kalman filter is used and the data fusion is carried out according to the optimal information fusion criterion with the minimum linear variance. Xu [12] proposed the local trajectory estimation, optimized ADS-B local trajectory estimation of TCAS and the optimized fusion trajectory estimations based on the data fusion algorithm of the Current Statistical (CS) model, however, it is not appropriate for civil aircraft when cruising.

Many scholars have proposed or improved the Interacting Multiple Model (IMM) and Variational Bayesian (VB) algorithm. A Sense-and-Avoid (SAA) system and IMM model is used to estimate the state of intruders by Ramsamy [13]. Sabatini and Chen [14,15] introduced a SAA system. An extended Kalman Filter and common IMM [16] algorithm were used to estimate the intruder's state vector. The IMM tracking models were focused on by the F.A.A. and the Thales Company. Constant Velocity (CV), Constant Acceleration (CA) and Constant Turning (CT) models are designed for surface tracking [17] and finally IMM was used to improve the accuracy of tracking. IMM has the characteristic of being adaptive, and can effectively adjust the probability of each model. However, few methods which are practical for real engineering issues. Variational Bayesian (VB) methods have been developed for a wide range of models to perform approximate posterior inference at low computational cost in comparison with the sampling methods (for a review of VB methods, see, e.g., [18,19]). The VB method was utilized by Sarkka [20] to propose an adaptive Kalman filtering method which is based on forming a separable variational approximation to the joint posterior distribution of states and noise parameters on each time step separately.

Due to the fact the small deviations in the previous simulation is not sufficient to prove the effectiveness of the fusion benefit, we built a fusion system based on a combined ADS-B physical system and TCAS simulation. The ADS-B minimum system is defined as a system with ADS-B equipment as the main part. It is in a semi-physical simulation environment which has some minor differences with the real system. An ADS-B and TCAS data fusion model based on an adaptive sampling period VB-IMM algorithm is proposed for integrated display from a theoretical model design, experimental simulation verification, engineering implementing the three parts in ADS-B In. An airspace traffic situation display is also presented. In ADS-B Out, some experimental verification processes include ADS-B Out data transmission based on a simulation cockpit platform and ADS-B Out data transmission based on an UAV platform.

The paper is organized as follows: Section 2 presents the design of ADS-B minimum system from two aspects of ADS-B In and Out including mathematical formulations and engineering implementation methods. Section 3 presents flight simulation and engineering implementation results of our ADS-B minimum system. Section 4 presents our conclusions and discusses future challenges and perspectives.

2. ADS-B Minimum System Design

The ADS-B minimum system design is shown from the two aspects of ADS-B In and Out. The processes of ADS-B In and Out are shown in Figure 1. The main theoretical design content of the ADS-B minimum system is a fusion model with a variable sampling VB-IMM (VSVB-IMM) algorithm. The part of engineering design and experimental verification includes airspace traffic situation display, ADS-B Out data transmission based on a simulation cockpit and UAV platform. The ADS-B minimum system design is shown in Figures 2 and 5.

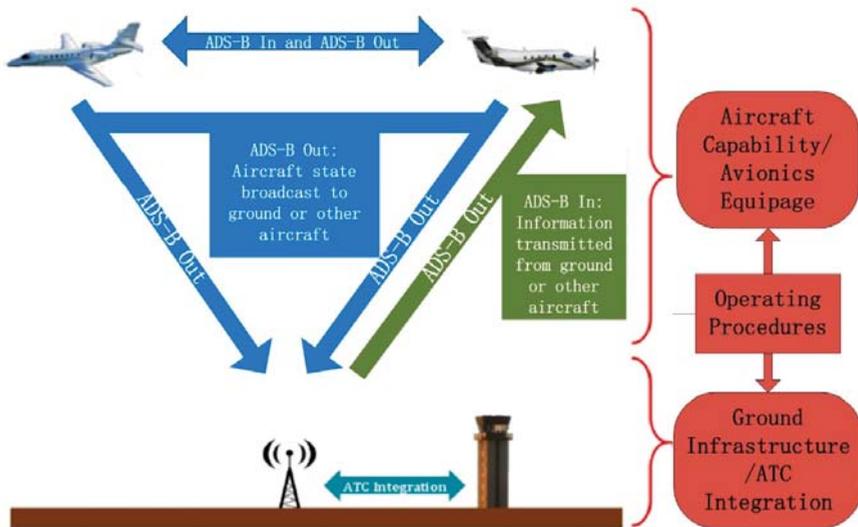


Figure 1. Automatic Dependent Surveillance–Broadcast (ADS-B) data transmission.

2.1. ADS-B In Minimum System Design

The steps of ADS-B In is that the broadcast data are received at an ADS-B station or ADS-B airborne equipment from 1090 MHz frequency and then the data is sent to the PC terminal through a serial port or network port for data processing. The integrated display and airspace traffic situation display is shown as the results of ADS-B In. The program of ADS-B In is shown in Figure 2.

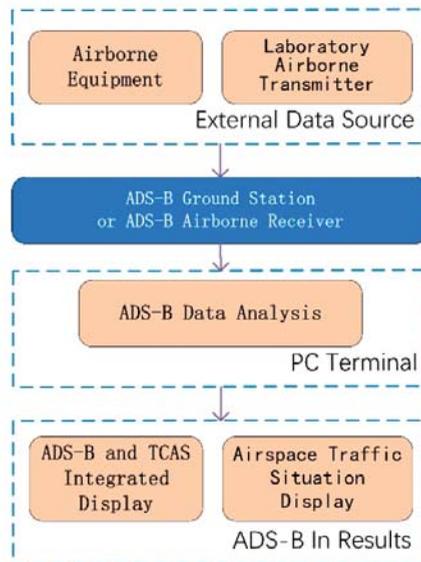


Figure 2. ADS-B In minimum system design.

2.1.1. ADS-B and TCAS Integrated Display Development

A VB-IMM algorithm was given by Dai [5] for maneuvering target tracking and estimating unknown noise variance. The 3-dimensional trajectory is generated by an aircraft movement model or flight simulation cockpit and IMM is utilized for filtering. The time-varying noise is estimated by the VB method, which is the basis for failure prediction and adjustment of the sampling period. The CV, CA and CS filter model is built in parallel in the IMM algorithm [21]. The state can be estimated by the filter weighting of different motion models. The local optimal value is obtained from variable sampling IMM (VS-IMM), fixed sampling IMM (FS-IMM) and the CS model algorithm. The global optimal value is obtained from the local optimal value following the optimal information fusion criterion and is used as input of the TCAS subsystem for obtaining the value of Tau (time until the closest point of approach) between aircraft. False alarms and leak alarms are counted for statistical analysis. The system frame diagram is shown in Figure 3.

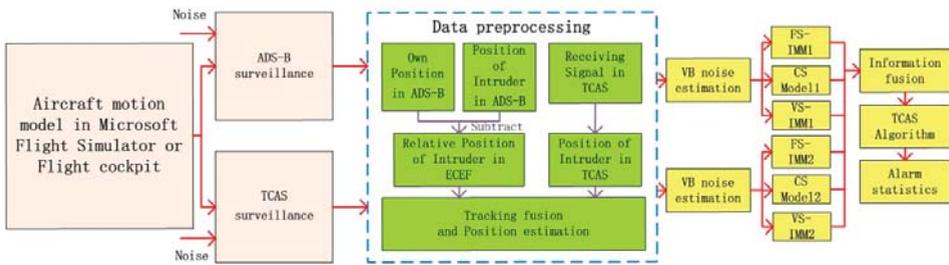


Figure 3. System frame diagram.

1. Data preprocessing

(1) Coordinate transformation

The ADS-B coordinate in WGS-84 and TCAS II data is the relative position. The coordinates of ADS-B and TCAS are transformed into a unified fusion coordinate system which is the “Earth-Centered, Earth-Fixed (ECEF)” coordinate system.

(2) Time synchronization

Although the update rate for ADS-B reports and TCAS II data is 1 s, the system delay will result in inconsistencies with data acquisition. Therefore, the data must be synchronized before data fusion. Since the ADS-B is based on GNSS, the ADS-B time base is used as a benchmark, so the measured values of TCAS II at time t_j are synchronized to the common processing time t_i of ADS-B which is shown in Equation (1):

$$Z_{TCAS II}(t_i) = Z_{TCAS II}(t_j) + v \times (t_i - t_j) \tag{1}$$

where $Z_{TCAS II}(t_i)$ is the measured values of TCAS II in time t_i . v is the velocity of aircraft and the $v \times (t_i - t_j)$ is a correction item.

2. Markov jump linear system and IMM model

The Markov jump linear system [22,23] is shown in Equations (2)–(5):

$$MX(k + 1) = \varnothing_j(k)X(k) + \omega(k) \tag{2}$$

$$Z(k) = H_jX(k) + v_j(k) \tag{3}$$

$$\omega(k + 1) = c\omega(k) + \zeta(k) \tag{4}$$

$$v_j(k + 1) = d_j v_j(k) + \eta_j(k) \tag{5}$$

where M is the singular square matrix which means $\det M = 0$. The system is regular which means $\det(\lambda M - \varnothing)$ is not constantly 0. The state vector $X(k)$ is an n -dimensional vector, the observation process $Z(k)$ is an m -dimensional vector and the subscript $j \in S = \{1, 2, \dots, s\}$ denotes the model. The matrix functions $\varnothing_j(k)$ and $H_j(k)$ are known. The vector $\zeta(k)$ is a zero-mean white Gaussian process noise and the $\eta_j(k)$ is an independent Gaussian measurement noise with zero-mean and variance to be estimated.

Applying the augmented state method to convert the colored process noise $\omega(k)$ into part of the system state. After the original system state component is expanded, the new system is defined as Equations (6) and (7):

$$\bar{X}(k + 1) = \bar{\varnothing}(k)\bar{X}(k) + \bar{\Gamma}\zeta(k) \tag{6}$$

$$Z(k) = \bar{H}_j\bar{X}(k) + \gamma_j(k) \tag{7}$$

where $\bar{X}(k) = \begin{bmatrix} X_1(k) \\ \omega(k) \end{bmatrix}$, $\bar{\varnothing} = \begin{bmatrix} L_1^{-1}T_1 & L_1^{-1}\Gamma_1 \\ 0 & c \end{bmatrix}$, $\bar{\Gamma} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. The colored process noise $\omega(k)$ converts to the white noise $\zeta(k)$. It exists orthogonal matrix P and Q to satisfy the equation $PMQ = \begin{bmatrix} L_1 & 0 \\ L_2 & 0 \end{bmatrix}$. L_1 is lower non-singular triangular matrix. $\bar{H}_j = H_j(\bar{\varnothing} - d_jI)$, $\gamma_j(k) = H_j\bar{\Gamma}\zeta(k) + \eta(k)$. The colored process noise $v_j(k)$ converts to the white noise $\gamma_j(k)$.

The paper describes the IMM algorithm with three models, including CV, CA and CS model. $X(k)$ is the state vector of target and it is a 3-dimensional vector consisting of position, velocity and acceleration which is described in Equation (8):

$$X(k) = [x \ \dot{x} \ \ddot{x}] \tag{8}$$

Zhou [24] proposed the CS model and processed the noise of maneuvering target acceleration with a Rayleigh distribution. The CS model is described in Equations (9)–(12):

$$X(k + 1) = \varnothing(k)X(k) + T(k)\bar{a} + \omega(k) \tag{9}$$

$$Z(k) = H_jX(k) + v_j(k) \tag{10}$$

$$\varnothing(k) = \begin{bmatrix} 1 & T & \frac{(-1+aT+e^{-aT})}{a^2} \\ 0 & 1 & \frac{(1-e^{-aT})}{a} \\ 0 & 0 & e^{-aT} \end{bmatrix}, T(k) = \begin{bmatrix} \frac{(-T+\frac{aT^2}{2}+\frac{(1-e^{-aT})}{a})}{a} \\ T - \frac{(1-e^{-aT})}{a} \\ 1 - e^{-aT} \end{bmatrix} \tag{11}$$

$$Q(k) = 2a\sigma_a^2Q \tag{12}$$

The process noise matrix $Q(k)$. Q is a constant matrix. σ_a^2 is the maneuvering acceleration variance. Maneuvering acceleration variance can be obtained in Equation (13):

$$\sigma_a^2 = \begin{cases} \frac{4-\pi}{\pi}(a_{max} - \hat{a}(k|k-1)), & \text{if } a \geq 0 \\ \frac{4-\pi}{\pi}(\hat{a}(k|k-1) - a_{-max}), & \text{if } a < 0 \end{cases} \tag{13}$$

where a_{max} and a_{-max} are the upper and lower acceleration limits and the acceleration estimation is $\hat{a}(k|k-1)$. The closer the real value of acceleration is to a_{max} , the smaller the σ_a^2 is. The closer the real value of acceleration is to zero, the larger the σ_a^2 is. The value of σ_a^2 affects the process noise $Q(k)$ which decides the tracking accuracy of maneuvering targets. The problem is how to make the system

have a constant speed or a small and large degree number of maneuvers track accurately as much as possible and scholars have put forward ideas for this:

$$f(x, y, \sigma) = 1 - e^{-\frac{(x-y)^2}{\mu\sigma^2}} \tag{14}$$

$$Q'(k) = Q(k) * f(x, y, \sigma) = 2a\sigma_a^2 Q * (1 - e^{-\frac{(x-y)^2}{\mu\sigma^2}}) \tag{15}$$

Li [25] used Gaussian membership function of fuzzy control theory to improve the CS model and the Equations (14) and (15) show the details of the improved model. The terms x, y, σ^2 and μ are the input of position, the estimation of current position, the variance of innovation and a constant, respectively. The value $|x - y|$ is big when there is a large maneuver, so $f(x, y, \sigma) \rightarrow 1$. Meanwhile $Q(k)$ is small and the improved model $Q'(k)$ is relatively small. On the contrary, a small maneuver causes $f(x, y, \sigma) \rightarrow 0$ and the value of $Q(k)$ is large, so it can also guarantee $Q'(k)$ is stable.

3. The formal algorithm flow

The formal algorithm flow is shown in Figure 4. The unknown measuring noise variance is estimated by the VB algorithm [26] which is in step one and two. m_k^- is the acceleration estimation and P_k^- is the a priori state covariance. The main equation is shown as follows:

$$m_k^- = \varnothing(k)m_{k-1}^- + T(k)m_{k-1}^- \tag{16}$$

$$P_k^- = \varnothing(k)P_{k-1}\varnothing(k)^T + Q(k) \tag{17}$$

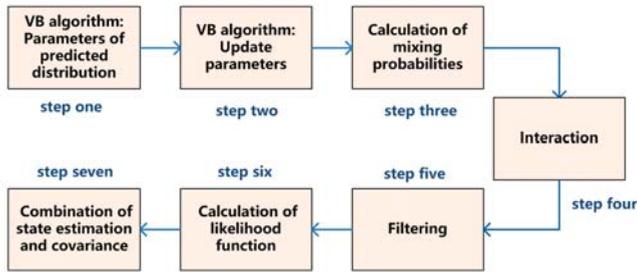


Figure 4. The formal algorithm flow.

The estimation of noise variance is input into the IMM model. The process of IMM filtering [27] is shown in steps three, four and five.

In step three, the process of calculation of mixing probabilities is described in Equation (18), where \bar{c}_j is a normalization factor:

$$\mu_{ij}(k-1|k-1) = \left(\frac{1}{\bar{c}_j}\right) p_{ij}\mu_i(k-1), \bar{c}_j = \sum_i p_{ij}\mu_i(k-1) \tag{18}$$

In step four, the system state is estimated which is calculated by a linear equation with $\hat{x}_i(k-1|k-1), \mu_{ij}(k-1|k-1), P_i(k-1|k-1)$, etc.

In step five, the measurement noise variance is obtained by VB estimation rather than a predetermined constant in the common IMM algorithm. The state estimation and covariance matrix are obtained by running each filter.

In step six, the likelihood function is calculated by the residual measurement and covariance. Finally, in step seven, $\hat{x}(k|k)$ and $\hat{p}(k|k)$ are estimated by a weighted sum of the estimations from all filters.

4. Variable sampling period VB-IMM [28]

The sampling period of VB-IMM is shown in Table 1. The F.A.A. issued the problem Airworthiness Approval of Automatic Dependent Surveillance-Broadcast Out systems in 2011 [29]. The concept of Navigation Accuracy Category for Position (NAC_p) specifies the accuracy of the aircraft's horizontal position information (latitude and longitude) transmitted from the aircraft's avionics. EPU is the Estimated Position Uncertainty. The sampling period is determined by position variance and NAC_p . The sampling period can increase appropriately when the variance estimated by the VB algorithm is relatively small, so the simulation sampling period is 1 s in the NAC_p 10 and 11 in Table 1. The sampling should be intensive when the estimated variance increases and meantime the intruder is closer. The sampling period in the simulation is 0.8 s in the NAC_p 9 and 0.6 s in the NAC_p 8, 7, 6. When the intruder is far away, the sampling period is restored to 1 s. The simulation sampling period is adaptive according to the value of NAC_p .

Table 1. Simulation sampling period corresponding to different Navigation Accuracy Category for Position (NAC_p) values.

NAC_p	Horizontal Accuracy Bound	Simulation Sampling Period
0	$EPU \geq 18.52$ km (10 nm)	1 s
1	$EPU < 18.52$ km (10 nm)	1 s
2	$EPU < 7.408$ km (4 nm)	1 s
3	$EPU < 7.408$ km (4 nm)	1 s
4	$EPU < 1852$ m (1 nm)	1 s
5	$EPU < 926$ m (0.5 nm)	0.8 s
6	$EPU < 926$ m (0.5 nm)	0.6 s
7	$EPU < 185.2$ m (0.1 nm)	0.6 s
8	$EPU < 92.6$ m (0.05 nm)	0.6 s
9	$EPU < 30$ m	0.8 s
10	$EPU < 10$ m	1 s
11	$EPU < 3$ m	1 s

5. Optimal Information Fusion Criterion

The unbiased estimations of L sensors, \hat{x} , $i = 1, \dots, L$. The estimation error covariance matrix P_{ij} , $i, j = 1, \dots, L$ is obtained. The fusion is performed according to a matrix weighted linear minimum variance criterion. There are two sensors (ADS-B and TCAS) in this paper, so $L = 2$. The process of fusion is given in Equations (19) and (20):

$$[A_1, \dots, A_L] = [A_1, A_2] = [A_{TCAS}, A_{ADS-B}] = (e^T P^{-1} e)^{-1} e^T P^{-1} \quad (19)$$

$$\hat{X}_f = \sum_{i=1}^L A_i \hat{x}_i = A_1 \hat{x}_1 + A_2 \hat{x}_2 = A_{TCAS} \hat{x}_{TCAS} + A_{ADS-B} \hat{x}_{ADS-B} \quad (20)$$

where A_i is an n -order square matrix, P is a block matrix with P_{ij} as the (i, j) element and $P_{ij} = cov(\hat{x}_i, \hat{x}_j)$, $i, j = 1, 2$. $e = [I_n, \dots, I_n]$, I_n is an n th-order identity matrix. We can hypothesize that the \hat{x}_{TCAS} and \hat{x}_{ADS-B} are unbiased because they are obtained from the results of variable sampling VB-IMM, fixed sampling VB-IMM and the CS model algorithm.

2.1.2. Airspace Traffic Situation Display Development

The source of data which is received by an ADS-B station or ADS-B airborne equipment is from the ADS-B signals of aircraft in the airspace. After decoding an ADS-B messages and transferring the data to the PC terminal, the flight information is obtained. The flight information includes longitude, latitude, height, ground speed, ground velocity direction, vertical velocity, etc. in plaintext. The flight information is displayed on the airspace traffic situation interface. The paper develops two airspace traffic situation interfaces according to different requirements for surveillance systems.

2.2. ADS-B Out Minimum System Design

ADS-B Out is designed from two aspects. The first is the ADS-B Out data transmission based on a simulation cockpit platform. The second is ADS-B Out data transmission based on an UAV platform. The program of ADS-B Out is shown in Figure 5.

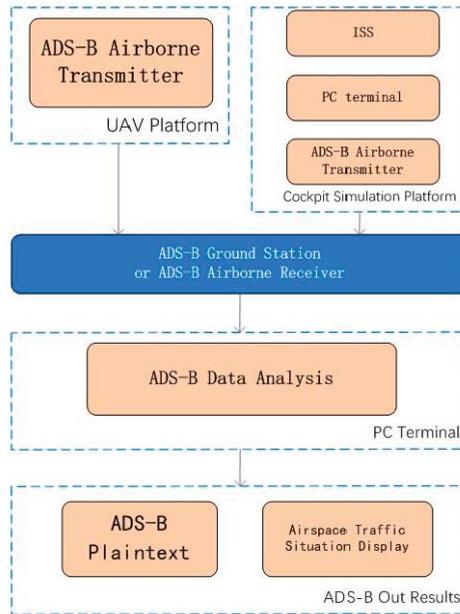


Figure 5. ADS-B Out minimum system design.

2.2.1. ADS-B Out Data Transmission Based on a Simulation Cockpit Platform

The simulation flight information is generated by the simulation cockpit platform. The flight information of the aircraft are sent to a PC terminal through an Integrated Surveillance System (ISS). After it is processed and converted by the protocols in the PC terminal, data are sent to the ADS-B airborne equipment. Then they are transmitted to the airspace. The ADS-B data is received by the ADS-B station or ADS-B airborne equipment. After decoding, the flight information can be obtained.

2.2.2. ADS-B Out Data Transmission Based on an UAV Platform

ADS-B airborne equipment is loaded on the UAV platform and it obtains the current flight information through the airborne equipment GPS and Beidou modules. The ADS-B signal is received by the ADS-B ground station or ADS-B airborne equipment, and the aircraft flight information is obtained after decoding. The UAV flight path is displayed and is compared with the flight path that is recorded by the positioning device of the UAV itself.

The main part of ADS-B minimum system is designed and shown in Figure 6. The three data source of this system include the flight simulation module, aircraft in airspace and the experimental UAV platform. After decoding, the plaintext information can be used in two ways. The first is that the plaintext information is sent to the display terminal to surveil the airspace traffic situation and detect any abnormal trajectory based on the recorded trajectory information or the flight recorder data. The second way is to send it to a Digital Mock-up Module to process the aircraft information

including information estimation and fusion, then into TCAS logic, to carry out a collision avoidance solution. Finally, all this information is displayed on a TCAS Digital Mock-up and cockpit platform.

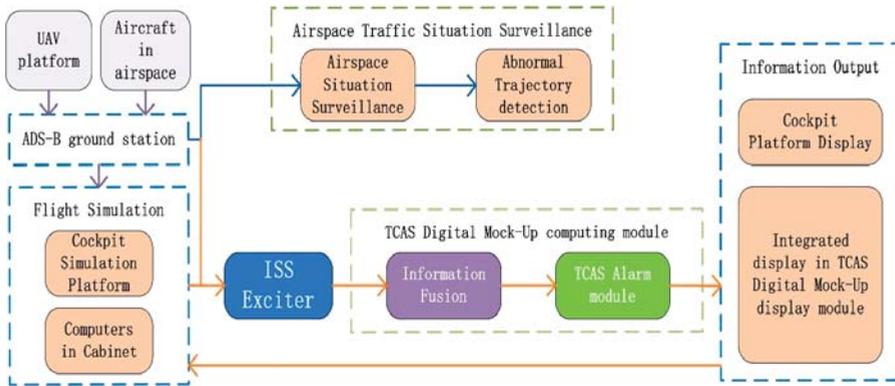


Figure 6. System design block diagram.

3. ADS-B Minimum System Implementation

3.1. ADS-B In Minimum System Implementation

3.1.1. ADS-B and TCAS Integrated Display

This scheme is the core of the ADS-B minimum system. According to the previous analysis of the ADS-B In minimum system program, this part shows the results of a VB-IMM algorithm simulation and application of an integrated display with the ADS-B and TCAS fusion system.

Simulation Results of Fusion Model Based on VB-IMM Algorithm

The 3-dimensional trajectory is generated by an aircraft movement model or flight simulation cockpit. The IMM model is utilized for filtering. The VB method is used to estimate the time-varying noise. The false alarm, leak alarm statistics and the data fusion's benefit analysis for the fused system are given to analyze the performance of system.

1. Flight path simulation

The parameters of simulation system are set as follows: Flight experience 3000 s, sampling period $T = 1$ s. Flight position: $98^{\circ}00'00''$ E, $29^{\circ}00'00''$ N, 4502 m height. It climbs from 300 m height and then cruises at constant height. The initial position of the intruder: $106^{\circ}00'00''$ E, $29^{\circ}00'00''$ N, 300 m height. The observation noise variance of TCAS is 50 dB, ADS-B's observation noise standard deviation is time-varying. The 3-dimensional trajectory is shown in Figure 7.

2. Variable sampling period VB-IMM (VSVB-IMM) and fixed sampling period VB-IMM (FSVB-IMM)

Noise is estimated by a Variational Bayesian. The observation noise variance of TCAS is 50 dB and the ADS-B's observation noise standard deviation is time-varying. In Figure 8, the variance of sinusoidal oscillation noise is 40 dB. The iteration is 30 in each cycle of this paper. Figure 8a,b are the online variance estimation of measurement noise for ADS-B and TCAS.

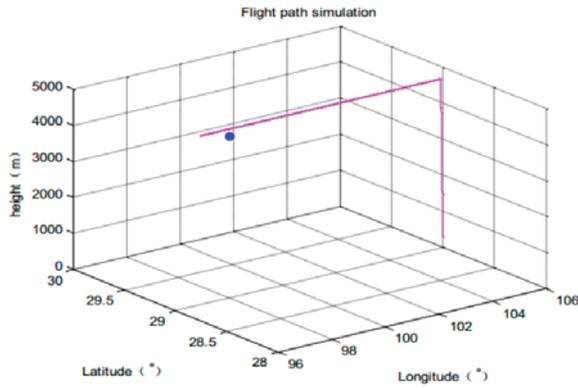


Figure 7. Aircraft trajectory.

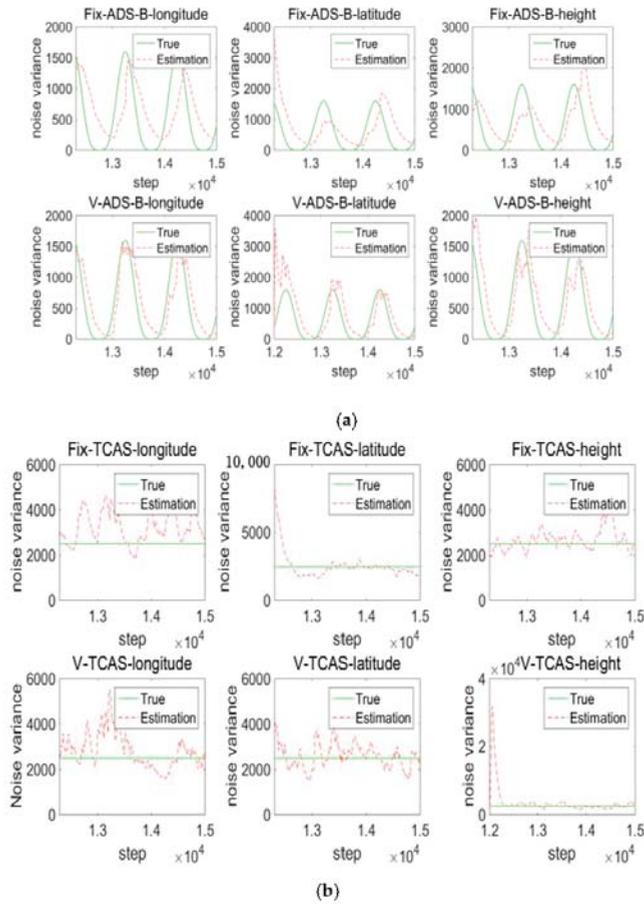


Figure 8. Noise estimation of system. (a) Noise estimation of ADS-B; (b) Noise estimation of Traffic Collision Avoidance System (TCAS).

The following is the statistical results of experiments. Combining Figure 9a with Figure 9b for analysis, we see the fused system’s estimation error is smaller than the TCAS and ADS-B subsystem’s estimation error. The fused track information is better than that of any subsystem. The fusion system is in a semi-physical simulation environment. A minimum system which combines the ADS-B physical system and TCAS simulation is built. The ADS-B measured values are obtained from an ADS-B Out experiment based on the simulator cockpit platform, which are sent to the air by the ADS-B airborne equipment, and received by an ADS-B ground station. Then the ADS-B measured noise is obtained from the ADS-B truth values and measured values.

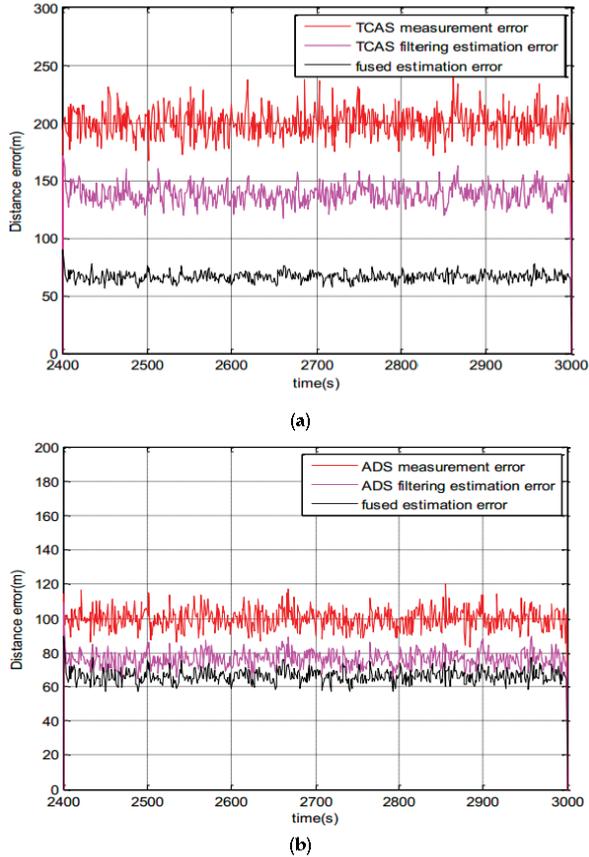


Figure 9. Measurement noise and fusion estimation noise. (a) Measurement noise and fusion estimation noise of TCAS; (b) measurement noise and fusion estimation noise of ADS-B.

3. Statistics of root mean squared error

The RMSE is calculated as Equation (21):

$$RMSE_k = \left[\frac{1}{M} \sum_i^M (x_k^{(i)} - \hat{x}_k^{(i)})^2 \right]^{\frac{1}{2}}, k = 1, 2, \dots, \text{step} \tag{21}$$

where M is the number of Monte Carlo iterations, the parameter k is the simulation step. The statistical result of 60 Monte Carlo experiments is shown in Figure 10.

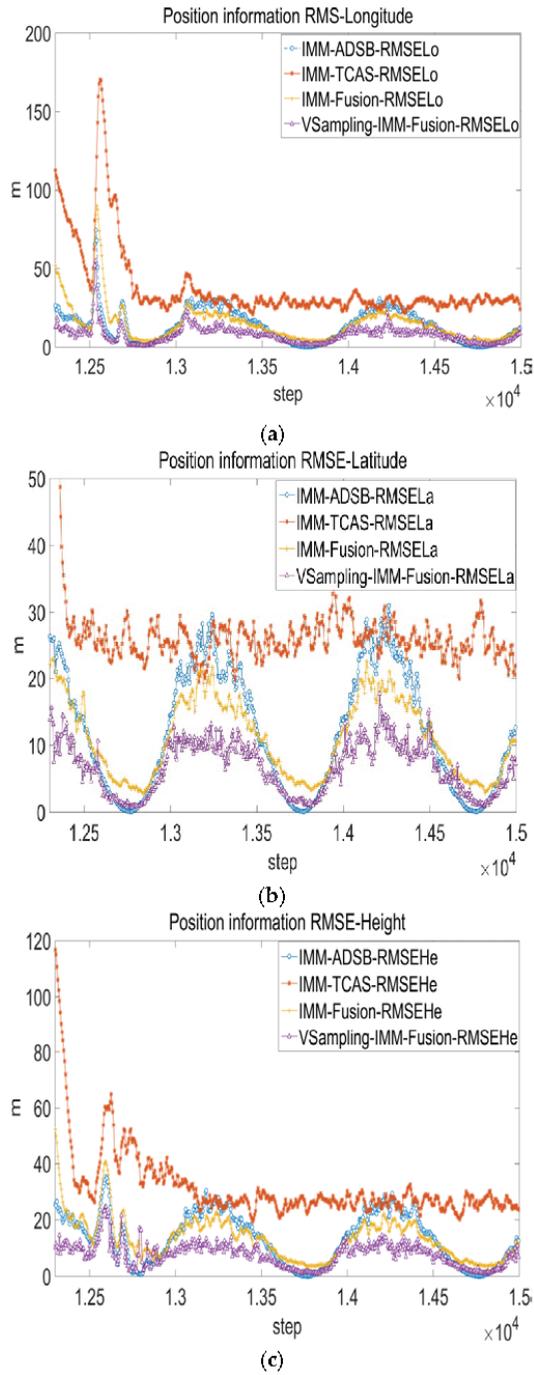


Figure 10. Root mean squared error of statistics. (a) Root mean squared error-Longitude; (b) Root mean squared error-Latitude; (c) Root mean squared error-Height.

The RMSE of fusion system is smaller than TCAS and ADS-B subsystem. The VSVB-IMM is superior to FSVB-IMM as purple line with triangles is the smallest most of the time.

4. TCAS CPA calculation

The CPA calculation results in large time period are shown in Figure 11a. The CPA calculation results in small time period are shown in Figure 11b. A fused track is added to the TCAS core solver model [30] to calculate the time of CPA. The trajectory is filtered by a VSVB-IMM model and the results of the fusion model are injected into the TCAS logic. The red line which represents the fused system is more close to the blue line which represents the CPA true value than any other subsystem.

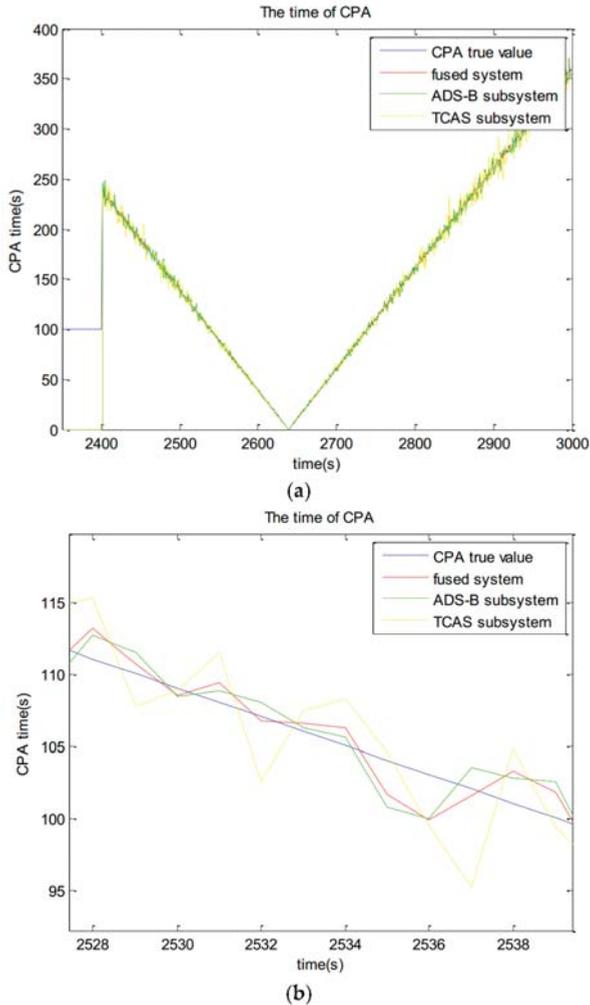


Figure 11. The CPA calculation results. (a) The CPA calculation results in large time period; (b) The CPA calculation results in small time period.

Figure 12 is a flowchart of the TCAS algorithm. This algorithm firstly generates track information by a flying solver model. The core processing program then receives data, conducts geodetic coordinate

system and coordinate system conversion, executes the CPA algorithms and calculates the relative position of aircraft. Finally, it is vital to estimate the time of encounter, anticipate conflicts and make RA decisions. The relative position of aircraft is calculated and the value of Tau is estimated based on trend extrapolation method. TA is the Traffic Advisories and RA is the Resolution Advisories in TCAS.

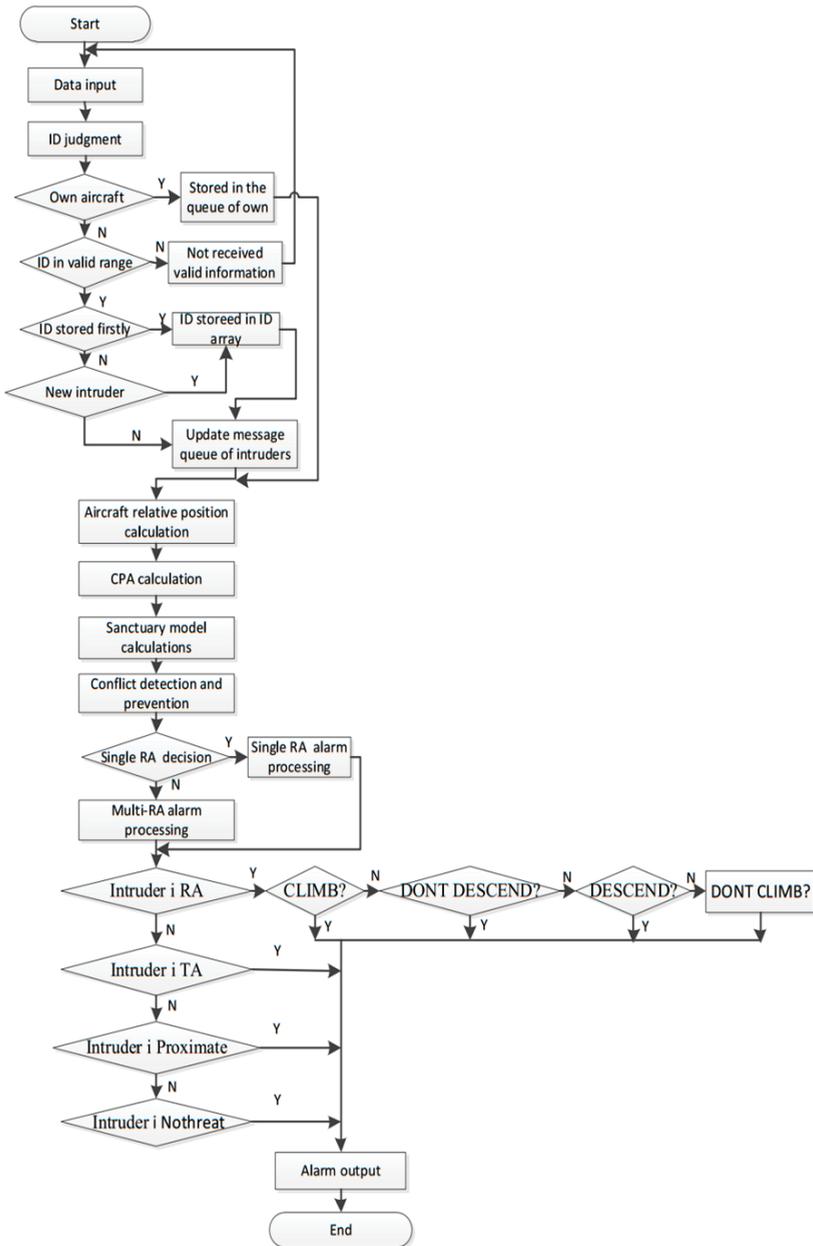


Figure 12. TCAS algorithm flowchart.

5. False alarm, leak alarm analysis of fused system

The experiment is repeated for 200 times independently. One of the statistics is the number of false alarms and leak alarms during the TA (CPA in 35–45 s) and RA (CPA < 35 s) period. Statistics for false alarms are when the actual time is ahead of the theoretical alarm time and exceeds the threshold value (it can be set to a constant 1 s). A leak alarm is when the actual time hysteresis of the theoretical alarm time exceeds the threshold value (1 s). In conjunction with Figure 13 and Table 2, qualitative and quantitative analysis can be carried out showing that the fused system can reduce the incidence of false and leak alarms in the TA and RA warning alarm interval. Leak alarms and delayed alarms compress the avoidance response’s time of the system and pilot, thus seriously affecting flight safety. Therefore, a more accurate alarm time can improve system security.

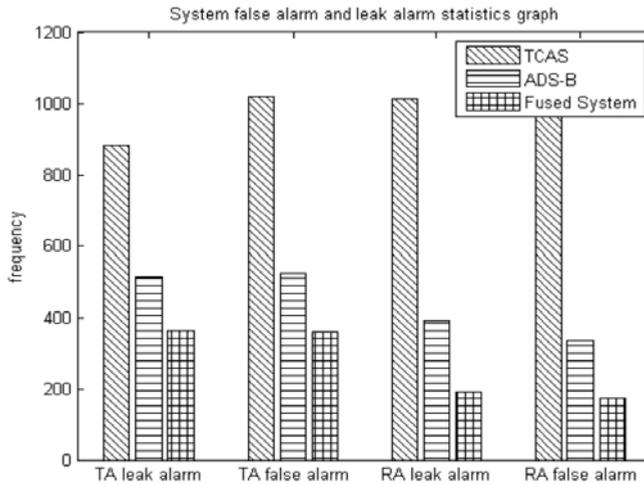


Figure 13. System false alarm and leak alarm statistics graph.

Table 2. Statistics of early alarm and hysteresis alarm during ta (35–45 s), ra (<35 s) in 200 experiments.

Alarm Type	System Categories		
	TCAS	ADS-B	Fused System
False alarm (TA) (frequency)	1018	524	361
Leak alarm (TA) (frequency)	883	513	362
False alarm (RA) (frequency)	1104	334	172
Leak alarm (RA) (frequency)	1014	390	192

6. CPA cumulative deviation analysis after RA decision

The TCAS core processor assesses the situation of the neighboring airspace and makes RA decisions. The intruder climbs to avoid detection at rate of 1500 feet per minute and meantime simulates the dynamic response where a human is in the loop. Pilot’s delay of practical response, mechanical and electrical system’s response delays can be taken into account. The delay was assumed to be a constant in this paper during the computer simulation. The simulation accumulated deviation data during the RA maneuver interval. It can be drawn from Figure 14 that the fused system can improve the cumulative deviation of CPA, which ensures a more precise response to ensure flight safety. Track graph of simulating maneuvering avoidance during RA is shown in Figure 15.

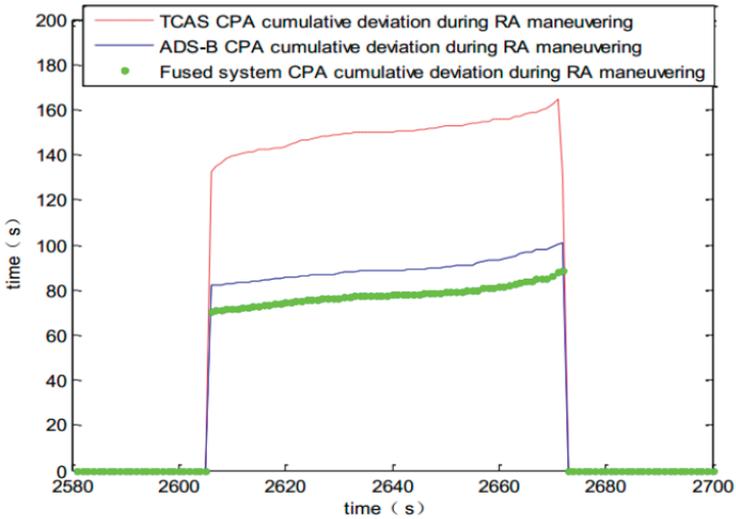


Figure 14. Accumulated deviation of CPA during the RA decision interval.

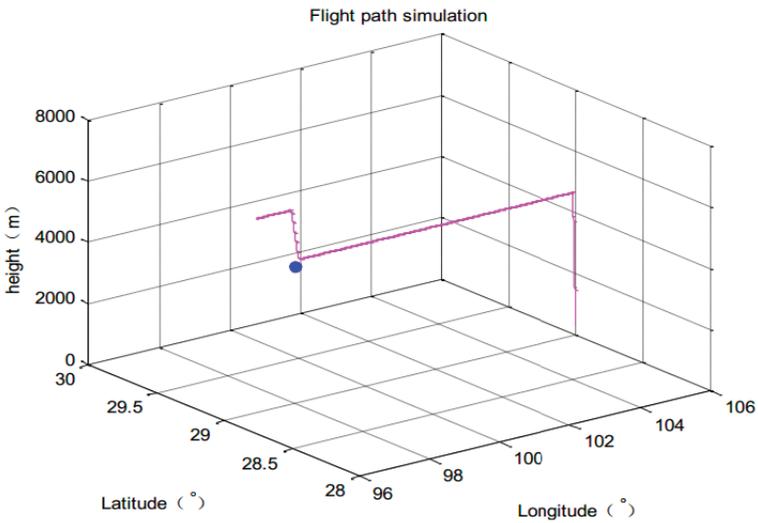


Figure 15. Track graph of simulating maneuvering avoidance during RA.

Application of ADS-B and TCAS Integrated Display

This part describes the engineering experiments and implementation of the ADS-B and TCAS data fusion algorithm. ADS-B message decoding, which obtains real-time flight information of the aircraft, is an important part of the ADS-B In. The main process of ADS-B message analysis is that ADS-B station or airborne equipment receives an ADS-B message and sends messages to the PC terminal through the network port. Then the message is decoded in the PC.

The data processing which includes calling the decoding function to decode the message, obtaining the message type and storing the corresponding variable is done. The message type has the air position information message (longitude, latitude and altitude in the air), the location information message (the longitude, the latitude, the running speed and the running speed direction of the ground target), the flight number information message (ICAO three word code, flight class set and flight category), speed information message (ground speed, ground speed and vertical speed) and error data messages.

Different ADS-B message types have different data composition order and data types. ADS-B message decoding program is that each data for independent analysis firstly and then the data analysis results are integrated, and finally the data statistics are calculated and stored.

The ADS-B message decoding program records the aircraft information received from the airspace. For each aircraft, the program updates the information and all of information is written in a file when a new ADS-B message is received and decoded. Table 3 is part of the ADS-B ground station to receive part of the aircraft information, receiving a location in the latitude 31°01′31.79″ N east longitude 121°26′29.75″ E near the range of about 300 km. The flow chart of the ADS-B message decoding software is shown in Figure 16.

Table 3. Part of aircraft information.

ID	Latitude	Longitude	Altitude	Speed North	Speed West	Speed Vertical
1	31.3268	122.629	4236.72	−45.2266	−312.967	−1728
2	31.3854	122.902	6156.96	−80.2507	−364.945	−1216
3	30.4038	121.241	4899.66	321.002	34.9823	64
4	30.6787	121.279	4038.6	351.002	36.9813	−1408
5	30.1474	121.154	5212.08	308.041	163.924	0
6	31.0127	122.763	7734.3	26.3675	477.98	1408
7	31.4885	123.445	7467.6	−76.2423	−351.948	64
8	29.8201	120.952	6454.14	355.048	191.911	−192
9	31.4381	123.173	7132.32	−82.2482	−362.944	−1088
10	31.0003	122.692	6156.96	0.335254	421	1792
11	30.2619	121.207	2834.64	263.014	87.9569	1088
12	31.7112	119.996	5394.96	−369.053	205.905	−2880
13	29.9003	121.456	3916.68	−13.2618	−336.99	1664
14	29.7201	122.295	8648.7	−94.3169	456.935	64
15	31.3854	122.902	6156.96	−80.2507	−364.945	−1216

The message decoding software process mainly includes three independent work modules: message reception, message decoding, processing and saving. An ADS-B message is transmitted in the link of UDP with a point-to-point protocol. The process of message analysis is shown in Figure 16 and the results of message processing and saving are shown in Table 3, which contains the discarding process for abnormal packets.

The integrated display of ADS-B and TCAS is based on message analysis. The ADS-B message received from the airspace by the ADS-B equipment is decoded by the PC terminal. The message is transmitted to the fusion system as an input in the simulator cockpit platform. The purpose of system design is to implement some functions including functionally testing the TCAS subsystem, improving the accuracy of TCAS prediction, reducing unnecessary alarms, and improving the effectiveness and safety of the system. The interface of the fusion system (TCAS Digital Mock-Up) in the laboratory is shown in Figure 16. If there is a possibility of aircraft traffic collisions happening, the alarm status will appear in the area.

TCAS digital Mock-Up, which is the platform of ADS-B and TCAS fusion system, can also handle single or multiple intrusion aircrafts including the four states of No threat, Proximate, TA and RA. The conditions for the system to work normally are that the integrated exciter software, integrated digital Mock-Up software and communication of all modules is normal. It combines the fusion model based on VSVB-IMM into the core algorithm which has been described above. The input of the fusion system (TCAS Digital Mock-Up) can be real-time flight data generated by an aircraft in the airspace or simulated flight data from the cockpit platform. The four situations including Nothreat, Proximate, TA and RA are shown in Figure 17a–d.

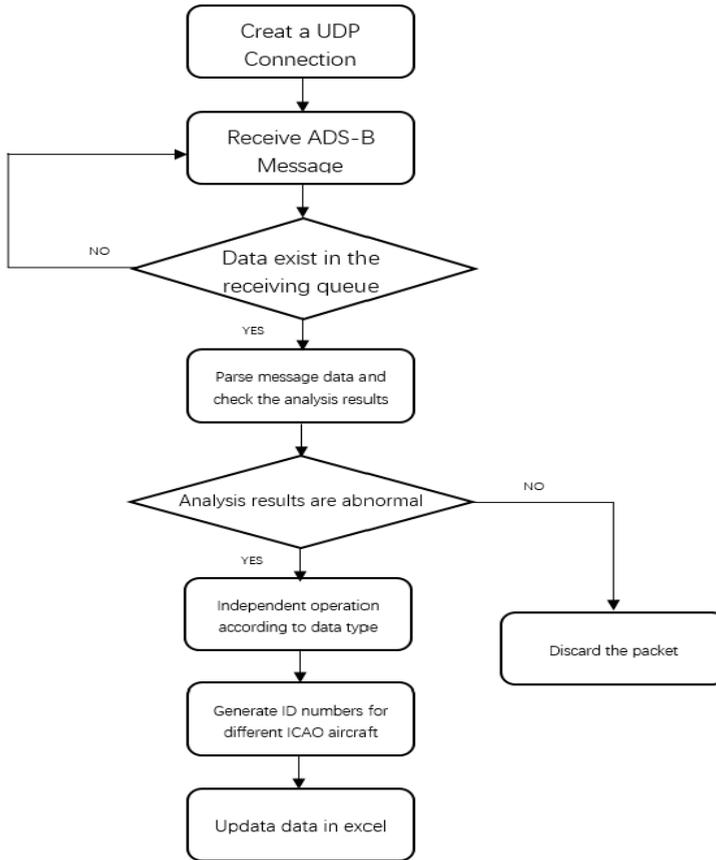


Figure 16. ADS-B message decoding software flow chart.

3.1.2. Display Interface Development

Figure 18 shows the airspace situation interface, which is able to display traffic conditions around 300 km in the air. The flight information which includes longitude, latitude, height, ground speed, ground speed direction, vertical speed and the type of aircraft is obtained by clicking on an aircraft icon. The flight path and height track information can also be more convenient to read out through the interface.

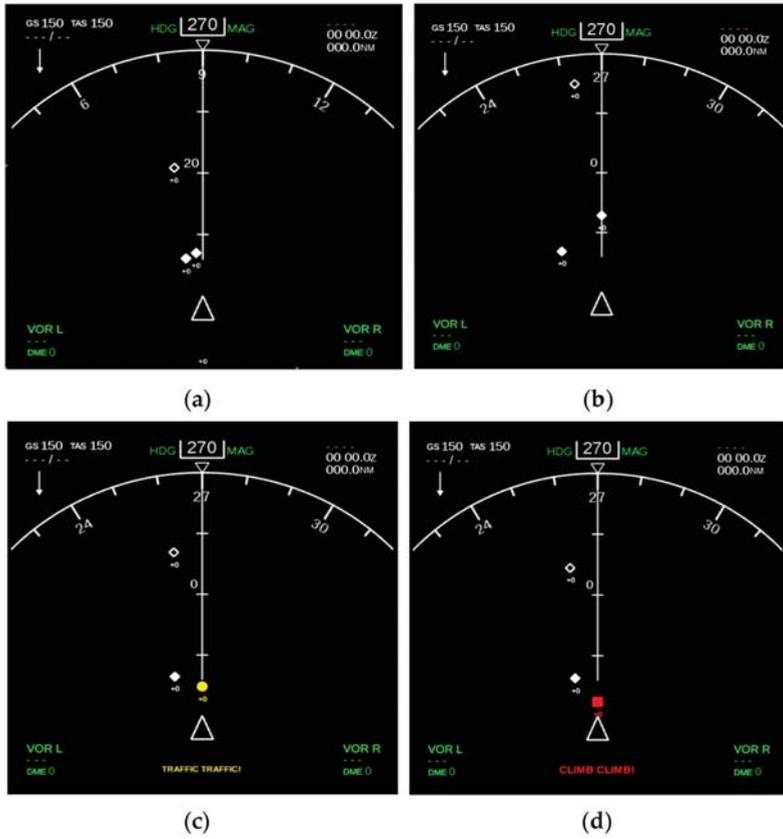


Figure 17. TCAS Digital Mock-Up. (a) Nothreat; (b) Proximate; (c) TA; (d) RA.

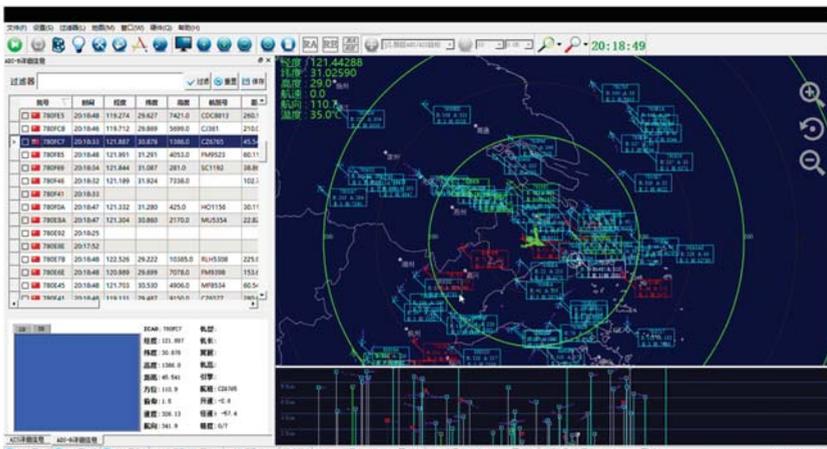


Figure 18. Airspace situation interface 1.

3.2. ADS-B Out Minimum System Implementation

3.2.1. ADS-B Out Data Transmission Based On Simulation Cockpit Platform

Simulated flight information which is generated by the simulation cockpit platform is transmitted to the airspace to simulate a real aircraft signal. Then the signal is received by the ADS-B station. Figure 19 shows the position of simulated aircraft and the signal received from the ADS-B station in the laboratory. The position of simulated aircraft is near the Shanghai Pudong Airport and over the airspace of the East China Sea. The process has benefits for experimental simulation work including traffic collision avoidance simulation, flight trajectory surveillance and verification, etc.

The airspace situation interface 2 is a new airspace situation interface in order to show some of the geographic information details. The interface can be modified to display 2D maps, satellite maps, or both by clicking on an icon. A variety of information including height trajectory of aircraft and other flight information is presented in it. The interface also supports two forms as input which are real-time flight data display and reads the saved data.



Figure 19. Airspace situation interface 2.

The development of the interface used a variety of Baidu map APIs to support the click control, slide the mouse wheel map zoom, click on the map switch, display the monitored aircraft location information and display the monitored aircraft height trajectory information. The interface implementation process is divided into four parts: create a map, set the map events, add controls to the map and add a cover to the map.

Compared with the monitoring interface shown in Figure 18, the airspace situation interface of Figure 19 which can be switched between 2D maps, satellite maps and mixed maps with clearer geographical features that can show more details of the map. The movement of the aircraft also shows a more intuitive and follow-up scene surveillance so research can continue on this basis.

3.2.2. ADS-B Out Data Transmit Based On UAV Platform

In this experiment, ADS-B airborne equipment was installed on the UAV platform to test ADS-B data transmission. The flight information is transmitted out by ADS-B airborne equipment and then received by an ADS-B station. Figure 20 is the flight trajectory with software on the UAV and the ADS-B signal.

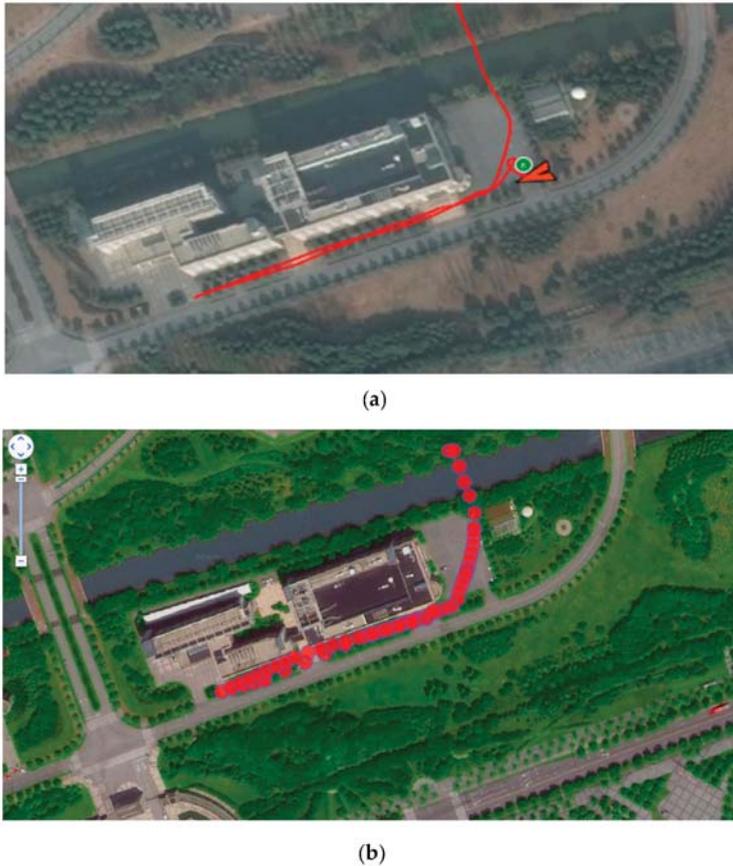


Figure 20. Unmanned Aerial Vehicle (UAV) trajectory verification. (a) Flight trajectory with software on the UAV; (b) Flight trajectory from ADS-B signal.

By comparison, the ADS-B signal emitted by the ADS-B airborne equipment mounted on the UAV platform is consistent with the recorded results of the UAV's own GPS flight track recorder. The experimental results show that the GPS module on the ADS-B airborne transmitter has a certain position accuracy and can meet the basic flight test and test requirements of the UAV platform.

4. Conclusions

Safeguarding human safety during aviation activities is the goal pursued by aviation practitioners and researchers. In our prior study, the simulation deviation of the fusion system is small but it is not sufficient to prove the effectiveness of the fusion benefit, so this paper built a minimal system based on an ADS-B physical system and TCAS simulation combined.

Based on the theory of ADS-B and TCAS, the paper designed and implemented an ADS-B minimum system from both theoretical and engineering aspects. A fusion model with a variable sampling VB-IMM (VSVB-IMM) algorithm is proposed in the theoretical aspect. A series of experimental simulations are done to verify the benefits of the fusion system including higher precision, lower false alarm rate and better flight security. In the engineering aspect, an airspace traffic situation

display, ADS-B Out data transmission based on a simulation cockpit platform and an UAV platform are designed and implemented from software architecture and comparative analysis design of the results.

More in-depth studies can be done in the future on the benefits of fusion system security and large data analysis of airspace surveillance. In time synchronization, some methods will be added to consider the measured time which is from the GPS navigation unit to transponder. The ADS-B and TCAS fusion system would be improved by exploring two-system fusion methods which meet the benefits and security of ADS-B and TCAS fusion. Pattern recognition or other methods can be used to identify the model which makes it possible to have traffic collision situations in the surveillance airspace and to alarm or perform some other action.

Acknowledgments: This paper are Sponsored by National Program on Key Basic Research Project (2014CB744903), National Natural Science Foundation of China (61673270), Shanghai Pujiang Program (16PJJD028), Aerospace Science and Technology Innovation Foundation (HTKJ/CX2015CAAA09), Shanghai Engineering Research Center of Civil Aircraft Flight Testing and Shanghai Science and Technology Committee Research Project (17DZ1204304).

Author Contributions: Yanran Wang conceived the main idea, designed the main algorithms and wrote the manuscript. Gang Xiao critically reviewed the paper, contributed to the revision of the paper and provided important suggestions. Zhouyun Dai built the simulation environment and performed the simulation experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aeronautics, R.T.C.F. *Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B)*; RTCA-DO-242A; RTCA Incorporated: Washington, DC, USA, 2006.
2. Aeronautics, R.T.C.F. *Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II)*; RTCA-DO-185B; RTCA Incorporated: Washington, DC, USA, 2009.
3. Kunzi, F. ADS-B Benefits to General Aviation and Barriers to Implementation. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2011.
4. Kuchar, J.; Drumm, A.C. The traffic alert and collision avoidance system. *Linc. Lab. J.* **2007**, *16*, 277.
5. Dai, Z.Y.; Xiao, G.; He, F.; Xie, J.H. An adaptive sampling VB-IMM based on ADS-B for TCAS Data fusion with benefit analysis. *J. Aeronaut. Astronaut. Aviat.* **2017**, *49*, 11–24.
6. Wang, X.G. The Design and Implementation of Air Traffic Control System Based on ADS-B Technology. Master's Thesis, University of Electronic Science and Technology of China, Chengdu, China, 2014.
7. Mccallie, D.; Butts, J.; Mills, R. Security analysis of the ADS-B implementation in the next generation air transportation system. *Int. J. Crit. Infrastruct. Prot.* **2011**, *4*, 78–87. [CrossRef]
8. Mueller, K.T.; Krozel, J. Aircraft ADS-B intent verification based on a Kalman tracking filter. In Proceedings of the AIAA Guidance, Navigation and Control Conference, Dever, CO, USA, 14–17 August 2000.
9. Peng, L.F.; Zhang, C.; Liu, Z.G.; Lin, Y.S. The CPR algorithm of 1090es automatic dependent surveillance—Broadcast (ADS-B) system. *J. Civ. Aviat. China* **2010**, *28*, 33–37.
10. Purton, L.; Abbass, H.; Alam, S. *Identification of ADS-B System Vulnerabilities and Threats*; Australian Transport Research Forum: Canberra, Australia, 2010; pp. 1–16.
11. Ni, Y.D.; Ma, Y.S.; Liu, P. Research on ADS-B and TCAS II data fusion algorithm. *Comput. Eng. Appl.* **2015**, *51*, 218–221.
12. Xu, X.J.; He, G.P. Data fusion algorithm of TCAS and ADS-B combined monitoring system based on current statistical model. *Telecommun. Technol.* **2013**, *53*, 1154–1157.
13. Ramasamy, S.; Sabatini, R.; Gardi, A. Avionics sensor fusion for small size unmanned aircraft sense-and-avoid. In Proceedings of the 2014 IEEE Metrology for Aerospace (MetroAeroSpace), Benevento, Italy, 29–30 May 2014; pp. 271–276.
14. Sabatini, R.; Gardi, A.; Ramasamy, S. A laser obstacle warning and avoidance system for unmanned aircraft sense-and-avoid. *Appl. Mech. Mater.* **2014**, *629*, 355. [CrossRef]
15. Chen, R.H.; Gevorkian, A.; Fung, A.; Chen, W.-Z.; Raska, V. Multi-sensor data integration for autonomous sense and avoid. In Proceedings of the AIAA Infotech@Aerospace Technical Conference, St. Louis, MO, USA, 29–31 March 2011.
16. Mazor, E.; Averbuch, A.; Bar-Shalom, Y.; Dayan, J. Interacting multiple model methods in target tracking: A survey. *IEEE Trans. Aerosp. Electron. Syst.* **1998**, *34*, 103–123. [CrossRef]

17. Lu, Y.; Liu, C.; Liu, P. An improved tracking method based on data mining in ADS-B for surface surveillance. In Proceedings of the 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Chongqing, China, 29–31 May 2012; pp. 1466–1470.
18. Jaakkola, T. 10 tutorial on variational approximation methods. In *Advanced Mean Field Methods: Theory and Practice*; MIT Press: Cambridge, MA, USA, 2001; p. 129.
19. Beal, M.J. Variational Algorithms for Approximate Bayesian Inference. Ph.D. Thesis, University of London, London, UK, 2003.
20. Sarkka, S.; Nummenmaa, A. Recursive noise adaptive kalman filtering by variational bayesian approximations. *IEEE Trans. Autom. Control* **2009**, *54*, 596–600. [CrossRef]
21. Jin, X.-B.; Du, J.-J.; Jia, B. Maneuvering target tracking by adaptive statistics model. *J. China Univ. Posts Telecommun.* **2013**, *20*, 108–114. [CrossRef]
22. Fu, X.; Jia, Y.; Du, J.; Yu, F. New interacting multiple model algorithms for the tracking of the manoeuvring target. *IET Control Theory Appl.* **2010**, *4*, 2184–2194. [CrossRef]
23. Song, G.; Jiang, S.; Lin, L. Information fusion Kalman filter with complex coloured noise for descriptor systems. *Chin. J. Sci. Instrum.* **2013**, *34*, 1195–1200.
24. Zhou, H. *Maneuvering Target Tracking*; National Defense Industry Press: Beijing, China, 1991; pp. 48–70.
25. Li, B.; Wang, Z. An improved target tracking algorithm based on the “current” statistical model. *J. Proj. Rockets Missiles Guid.* **2008**, *2*, 026.
26. Mbalawata, I.S.; Särkkä, S.; Vihola, M.; Haario, H. Adaptive metropolis algorithm using variational bayesian adaptive kalman filter. *Comput. Stat. Data Anal.* **2015**, *83*, 101–115. [CrossRef]
27. Bar-Shalom, Y.; Challa, S.; Blom, H.A. Imm estimator versus optimal estimator for hybrid systems. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 986–991. [CrossRef]
28. Mallick, M.; La Scala, B.F. IMM estimator for ground target tracking with variable measurement sampling intervals. In Proceedings of the 2006 9th International Conference on Information Fusion, Florence, Italy, 10–13 July 2006; pp. 1–8.
29. Hempe, D. *Airworthiness Approval of Automatic Dependent Surveillance-Broadcast (ADS-B) Out Systems*; AC-20-165B; US Department of Transportation Federal Aviation Administration: Washington, DC, USA, 2010.
30. Gu, B. The Simulation Algorithm of Civil Aircraft’s Air Traffic Collision Avoidance System. Master’s Thesis, Shanghai Jiaotong University, Shanghai, China, 2012; pp. 41–44.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Model-Based Heterogeneous Data Fusion for Reliable Force Estimation in Dynamic Structures under Uncertainties

Babak Khodabandeloo, Dyan Melvin and Hongki Jo *

Civil Engineering and Engineering Mechanics, The University of Arizona, Tucson, AZ 85721, USA; Khodabandeloo.babak@gmail.com (B.K.); dyanmelvin@email.arizona.edu (D.M.)

* Correspondence: hjo@email.arizona.edu; Tel.: +1-520-621-6589

Received: 31 August 2017; Accepted: 6 November 2017; Published: 17 November 2017

Abstract: Direct measurements of external forces acting on a structure are infeasible in many cases. The Augmented Kalman Filter (AKF) has several attractive features that can be utilized to solve the inverse problem of identifying applied forces, as it requires the dynamic model and the measured responses of structure at only a few locations. But, the AKF intrinsically suffers from numerical instabilities when accelerations, which are the most common response measurements in structural dynamics, are the only measured responses. Although displacement measurements can be used to overcome the instability issue, the absolute displacement measurements are challenging and expensive for full-scale dynamic structures. In this paper, a reliable model-based data fusion approach to reconstruct dynamic forces applied to structures using heterogeneous structural measurements (i.e., strains and accelerations) in combination with AKF is investigated. The way of incorporating multi-sensor measurements in the AKF is formulated. Then the formulation is implemented and validated through numerical examples considering possible uncertainties in numerical modeling and sensor measurement. A planar truss example was chosen to clearly explain the formulation, while the method and formulation are applicable to other structures as well.

Keywords: force estimation; heterogeneous sensor network; Kalman filtering; multi-metric measurements; structural dynamics

1. Introduction

Many engineering structures are subjected to various natural and man-made dynamic loads, including wind, earthquake, traffic, machine vibrations, and tidal loads. The structures can be damaged severely when the applied loads are stronger than the structural capacities. Even for seemingly small level loads, their continuing and long-term application may cause gradual degradation of the structural performances over time, such as fatigue problems, so identifying the time histories of applied forces to structures can significantly improve their design and effectively protect them against damaging loading events. However, direct measurements of applied forces are impossible in practice or very difficult and costly for dynamic structures. Even if structural responses (e.g., acceleration, velocity, displacement, or strain) are measured at the locations where the forces are applied, capturing the external forces from the measured responses is not possible. Even in the case that transducers are available for force measurements, when the structure is large such as long span bridges, towers, and skyscrapers, or when the spatial distribution of loading is complex, covering the entire structure at the locations where forces would be applied is a challenging task. Therefore, indirect methods for force identification have been investigated to overcome such limitations [1–3].

Input force identification is the procedure to determine the loads applied to the structure using its measured responses. In other words, the goal is to find the inputs to the system using the known

or measured outputs. Theoretically, having the frequency response functions (FRF) matrix together with the measured responses, it is possible to estimate the applied forces by inverting the FRF matrix. However, a unique solution is usually not available, because of the rank deficiency in the FRF matrix. The FRF-based input estimation is an ill-posed problem in general, so the presence of noise and small deviations will cause significant errors that are far from reality [4–7]. To overcome this issue for FRF-based methods, additional information, e.g., spatial distribution of loads, must be needed to have a unique solution [8,9]. Classical time and frequency domain methods for input identification also suffer from the requirement of needing an exact model, which is not possible in practice [10].

A variety of the load reconstruction approaches have been developed for specific force patterns; including impulsive load estimation for composite structures [6,11], harmonic force estimation in rotating machinery [12], and moving load identification in bridge and railway systems [13,14]. A sum of weighted accelerations technique (SWAT) is a well-known method for reconstructing impulsive loads from measured responses. In this method, measured accelerations are scaled by effective weights, which are the coefficients of equivalent mass at each acceleration location, to estimate the applied loading. This method is only suitable for systems with free boundary conditions [15,16]. For impact estimation of nonlinear structures, artificial neural network-based methods are known to improve the quality of the load estimation [17,18]. Such methods, unlike those based on convolution relation, can construct the nonlinear relationship between inputs and outputs. However, the mathematical model and algorithm used in this method has to be trained carefully by applying known forces to the structure; which is a critical limitation when applying it to large structures that have complex load patterns and distributions. A Bayesian inference-based regularization method has been proposed to identify the excitation forces [19]; they have estimated the low frequency components with good accuracy but the results for higher frequencies are not given.

In contrast to the force identification methods in frequency domain methods based on the fast Fourier transform, time domain methods were developed [20–22]. These methods in general consist of two main steps. First, using an operation matrix, the sequences of inputs are mapped to the outputs. In the second step, ill-posed inverse problem is solved with a regularization method. However when the size of operation matrix increases the force identification becomes highly difficult to solve. To overcome the problem of large operation matrix, sequential deconvolution input reconstruction (SDR) method was proposed [23]. Extensive parametric study of sequential deconvolution for input reconstruction was conducted [24] and it was concluded that the method can identify the inputs accurately in a moderately noisy environment.

Techniques based of the Kalman Filtering (KF) method have proven to be effective in identification of different types of loadings [25]. As a recursive linear state estimator, the Kalman Filter can provide statistically optimal estimates of the state, even with uncertainties in modeling and measurements, in the sense that the error covariance matrix is minimized. It has application in many areas, including navigation, object tracking, economics, signal processing, etc. [26]. The Kalman Filter was initially developed for linear systems. But if states are described by nonlinear equations or the observation relation is nonlinear, the extended Kalman Filter approaches have been explored [27–29]. To accommodate the non-gaussian noise and nonlinear dynamical system, Ensemble Kalman Filter (EnKF), which can be considered as a variant of interacting particle systems or particle filtering, has been developed using non-parametric approach [30]. However, in certain cases EnKF has poor performance [31,32]. For example, EnKF has difficulties to handle highly non-Gaussian posterior pdfs [33].

The Kalman filtering variants have been used for response identification at unmeasured locations [34–36], parameter identification, and damage detection [29]. A Kalman Filter approach in conjunction with a recursive least-square algorithm has been developed for force excitation estimations [6,25]. But this method requires data vectors that contain displacement measurements at all degree of freedoms which is not possible in most of the practical applications. Deconvolution Kalman Filter (DKF) is recently introduced and estimates the input forces using an augmented state

space system and Kalman filtering [37]. Augmented state space system is formed from combination of state space matrices from an auto regressive moving average (ARMA) and the mechanical system response modeling. Kalman filter-based approach is used to estimate external forces and torques for a serial-chain robotic manipulator [38]. In this case, dynamic model of the robot and available motor signals such as current, angles, and speed are used and there is no need for additional sensing. In lead-through programming (LTP), which is a fast approach for teaching a trajectory by physical interaction with robot, it is possible to use a sensor-less method based on Kalman filter using the generalized momentum formulation [39]. The quality of estimated torque relies on dynamic model of robot and the friction model.

Another variant of force identification using Kalman filter is to incorporate the unknown forces into state vectors and estimate the unknown forces as part of the states; this method is called Augmented Kalman Filter (AKF) [40]. In the original approach to use the AKF, acceleration responses are used in the measurement update of the Kalman Filter for force estimation. Stability issues of the AKF method have been investigated when accelerations are the only measured responses [41]. Because the error covariance matrix of AKF has the simple form of Riccati equations, it is discussed based on analytical arguments that estimations based solely on acceleration measurement are inherently unstable. To overcome the instability of input estimation from measured accelerations, dummy measurement method is suggested [41]. But if there are slowly varying low-frequency forces, the method will not be able to trace them. The other source of numerical instabilities in force estimation is when there are more sensors than the order of reduced-order dynamic model; reduced order models are frequently used since limited number of vibration modes dominate the response of a structure subjected to dynamic loading [37]. Furthermore, joint input-state estimation was applied in structural dynamics response and input estimation when accelerations are measured [42] and was further extended to account for unknown stochastic excitations [43]. To resolve the numerical issues related to the rank deficiency and un-observability of the AKF, dual Kalman filter approach for joint input-state estimation was proposed [44] and experimentally validated [45]. The mentioned limitations of AKF are solved through successive structure of dual Kalman filter.

Acceleration responses are usually the easiest and cheapest to measure and have been widely used in structural system identification and response and force estimations. However, their performance is not satisfactory in the low frequency range [46]. On the other hand, strain measurements work perfectly at low frequencies and show direct relations to stress, hence failure and fatigue [47]. Force identification relying solely on acceleration response measurements suffer from deficiencies such as instability, inaccuracy, and even possibly misleading results. Incorporation of displacement measurement is one remedy for the problem but it is expensive and complicated.

In this paper, a multi-metric approach is investigated to improve the stability and accuracy of the force estimation using the AKF method. It is shown how the stability issue of the AKF can be addressed and force estimation accuracy in both low- and high-frequency range can be enhanced by combined use of multi-metric measurements, i.e., strain and acceleration responses measured at limited locations of the structure, in the measurement update stage of the Kalman filter. The efficacy of the proposed method is numerically validated using a planar truss bridge structure model. The force estimation performances of the multi-metric method are compared with the applied reference loads under various modeling and loading conditions, considering nonzero-mean loads, and measurement noises. The proposed method is schematically shown in Figure 1.

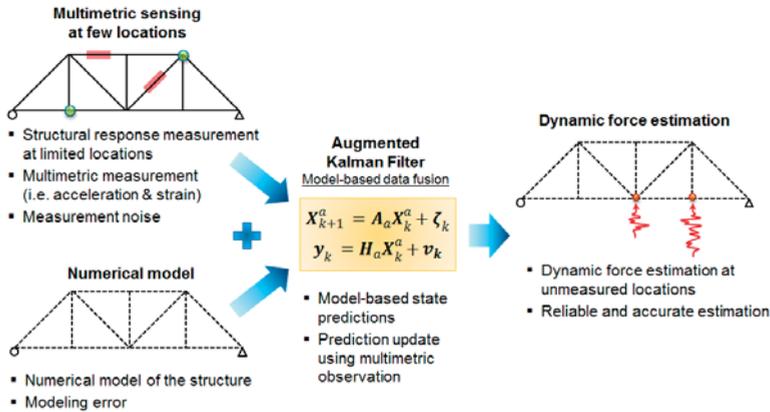


Figure 1. Schematic of proposed model-based heterogeneous data fusion for reliable force estimation.

2. Augmented Kalman Filter A

As a recursive linear state estimator, the Kalman filter provides statistically optimal estimates of the states in a way that minimizes the mean of the squared error, assuming the system behaves linearly and potential process noise and measurement noise are zero-mean Gaussian stochastic process. The states are predicted using a system (i.e., numerical model of a structure) and the predictions are updated using observations in a minimum-variance unbiased sense. This section describes the conventional Kalman filter for response estimation and its extended approach, i.e., augmented Kalman filter (AKF), for estimating both structural responses and input forces [40]. Furthermore, the stability issue of the AKF is discussed.

2.1. State Space Model

Dynamic behavior of a linear mechanical system is described by a second order differential equation of motion:

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = S_f f(t), \tag{1}$$

where $M_{n \times n}$, $C_{n \times n}$, and $K_{n \times n}$ are mass, damping, and stiffness matrices and $S_{f_{n \times n_f}}$ is a force selection matrix. “n” is the number of degrees of freedom and “ $n_f \leq n$ ” is the number of degrees of freedom which the force is applied to. The state space formulation of the above equation of motion in continuous domain is [48]:

$$\dot{x}(t) = A_c x(t) + B_c f(t), \tag{2}$$

where the subscript “c” stands for the continuous case:

$$x(t) = \begin{Bmatrix} u(t) \\ \dot{u}(t) \end{Bmatrix}; A_c = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}; B_c = \begin{bmatrix} 0 \\ M^{-1}S_f \end{bmatrix},$$

in the above equations $x_{2n \times 1}$ is the state vector. A set of equations, in addition to those describing the dynamics of the system, defined to describe the measured or observed values in term of the system states:

$$y(t) = H_c x(t) + D_c f(t), \tag{3}$$

the state Equation (2) together with the measurement Equation (3) form the continuous state space model of the system.

The discrete time form of the state space model is defined as:

$$x_{k+1} = Ax_k + Bf_k, \quad (4)$$

$$y_k = Hx_k + Df_k, \quad (5)$$

where:

$$A = e^{A_c \Delta t}, \quad (6)$$

$$B = \int_0^{\Delta t} e^{A_c \tau} d\tau B_c = A_c^{-1}(A - I)B_c \quad (7)$$

$$H = H_c, D = D_c. \quad (8)$$

The Kalman filter estimates the states in a recursive optimal manner from the state space formulation considering presence of uncertainties, either process errors (w_k) or measurement noises (v_k):

$$x_{k+1} = Ax_k + Bf_k + w_k \quad (9)$$

$$y_k = Hx_k + Df_k + v_k \quad (10)$$

2.2. Kalman Filter

Kalman filter has two main steps, i.e., time update and measurement update equations [27]:

Time update:

$$\begin{aligned} \hat{x}_{k+1|k} &= A\hat{x}_{k|k} + Bf_{k-1} \\ P_{k+1|k} &= AP_{k|k}A^T + Q, \end{aligned} \quad (11)$$

Measurement update:

$$\begin{aligned} K_{k|k} &= P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_{k|k}(y_{k|k} - H\hat{x}_{k|k-1}) \\ P_{k|k} &= (I - K_{k|k}H)P_{k|k-1}, \end{aligned} \quad (12)$$

where, Q and R are process (modeling) error and measurement noise covariance matrices, which are assumed to be known, constant, and independent of time and represented by $E\{w_i w_j^T\} = Q\delta_{ij}$ and $E\{v_i v_j^T\} = R\delta_{ij}$, respectively; where δ_{ij} is the Kronecker delta. For given covariance matrices Q_c and R_c for continuous system, the discrete Q and R can be obtained as below:

$$\begin{aligned} Q &= \int_0^{\Delta t} e^{A_c \tau} Q_c e^{A_c^T \tau} d\tau \\ R &= R_c / \Delta t \end{aligned} \quad (13)$$

and their order of magnitude is determined by the order of magnitude of state vector and signal to noise ratio of sensors [40]. For more detail on the terms used in Kalman filter, see [27].

2.3. Augmented Kalman Filter (AKF)

Conventional formulation of the Kalman filter described above in (11) and (12) can provide optimized estimates of the structural responses as the states (i.e., displacements and velocities) with given information of input forces (f). However, the input forces are rarely known in reality, so additional methods are needed to estimate the input forces [6,25].

The Augmented Kalman Filter (AKF) approaches use the augmented state vector that includes the input force, and then estimates structural responses as well as the input forces together as the part of the states. In order to build the full system equations for the AKF, an input force model should be

required. A random walk model is a general approach for the input force model, the state equation for the random walk model in continuous domain is defined:

$$\dot{f} = 0 + \eta, \tag{14}$$

where η is the input force model noise on the derivative of the force parameter, implying that the force derivative or force increment is a completely random process.

Its discrete version is known as a Martingale process as below:

$$f_{k+1} = f_k + \eta_k \tag{15}$$

The original state vector is then extended to include the input force. Augmented state equation in continuous time form is:

$$\begin{Bmatrix} \dot{x}(t) \\ \dot{f}(t) \end{Bmatrix} = A_{ac} \begin{Bmatrix} x(t) \\ f(t) \end{Bmatrix} + \begin{Bmatrix} w \\ \eta \end{Bmatrix} \tag{16}$$

$$A_{ac} = \begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix} \tag{17}$$

In the above equation, A_{ac} , is system matrix for augmented formulation in continuous form. Observation equation is:

$$y = H_{ac} \begin{Bmatrix} x(t) \\ f(t) \end{Bmatrix} + v \tag{18}$$

$$H_{ac} = [H \ D] \tag{19}$$

In discrete form, the augmented state vector and the state equation are:

$$X_k^a = \begin{Bmatrix} X_k \\ f_k \end{Bmatrix}_{(n_s+n_p) \times 1} \tag{20}$$

In the above equation n_s and n_p are number of states and inputs, respectively:

$$X_{k+1}^a = A_a X_k^a + \zeta_k \tag{21}$$

$$A_a = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}$$

Observation equation becomes:

$$y_k = H_a X_k^a + v_k \tag{22}$$

$$H_a = [H \ D] \tag{23}$$

matrices H and D are given in Equation (30).

Then the time and measurement update equations in the KF method become [40]:

Measurement update:

$$L_k = P_{k|k-1} H_a^T (H_a P_{k|k-1} H_a^T + R)^{-1} \tag{24}$$

$$\hat{X}_{k|k}^a = \hat{X}_{k|k-1}^a + L_k (y_k - H_a \hat{X}_{k|k-1}^a)$$

$$P_{k|k} = P_{k|k-1} - L_k H_a P_{k|k-1}$$

Time update:

$$\hat{X}_{k+1|k}^a = A_a \hat{X}_{k|k}^a \tag{25}$$

$$P_{k+1|k} = A_a P_{k|k} A_a^T + Q_a$$

In augmented Kalman filtering, since the force vector is included in state vector, the modeling error covariance matrix Q , together with regularization matrix (S) which is the covariance matrix of input noise, η , form the augmented covariance matrix Q_a :

$$Q_a = \begin{bmatrix} Q & 0 \\ 0 & S \end{bmatrix} \tag{26}$$

To elaborate the observation (measurement) matrix H_{ac} in augmented state space form, shown in the Equation (22), the following general relation between the measurement and structural responses is considered:

$$y(t) = S_a \ddot{u}(t) + S_v \dot{u}(t) + S_d u(t) \tag{27}$$

S_a , S_v , and S_d are " $n_d \times n$ " selection matrices corresponding to acceleration, velocity, and displacement state, respectively. " n_d " is the number of measurements and " n " is the number of degrees of freedoms.

We have:

$$\begin{aligned} \ddot{u}(t) &= -M^{-1}C \dot{u}(t) - M^{-1}Ku(t) + M^{-1}S_f f(t) \\ y(t) &= S_a \left\{ -M^{-1}C \dot{u}(t) - M^{-1}Ku(t) + M^{-1}S_f f(t) \right\} + S_v \dot{u}(t) + S_d u(t) \\ y(t) &= (S_v - S_a M^{-1}C) \dot{u}(t) + (S_d - S_a M^{-1}K) u(t) + S_a M^{-1} S_f f(t) \\ \Rightarrow H = H_c &= [S_d - S_a M^{-1}K, S_v - S_a M^{-1}C], D = D_c = S_a M^{-1} S_f \end{aligned} \tag{28}$$

When strain is measured, the strains can be formulated as the linear combination of displacement states, then the general relation Equation (27) can be rewritten as:

$$y(t) = S_a \ddot{u}(t) + S_v \dot{u}(t) + S_d u(t) + S_s u(t) \tag{29}$$

where S_s is the strain selection matrix. Therefore, if strain measurements are considered, then (28) becomes:

$$H = H_c = [S_s + S_d - S_a M^{-1}K, S_v - S_a M^{-1}C], D = D_c = S_a M^{-1} S_f \tag{30}$$

2.4. AKF Update via Multi-Metric Observation

Multi-metric approaches combining specialized metrics have tremendous potential to enhance the quality of the obtained information, providing a comprehensive way to take the respective advantages and to overcome the weakness of such single-metric methods [49,50]. Civil structures are exposed to both low- and high-frequency force excitations; therefore, the Multi-metric approach investigated here can improve the AKF accuracy for estimating broadband force excitations.

In addition, the Multi-metric method can contribute to solving the stability issue of the AKF. Naets et al., (2015) analytically pointed out the stability problem of the AKF using the Popov-Belevitch-Hautus (PBH) criterion. Particularly, when acceleration measurements are only used in the measurement update, the AKF intrinsically suffers from the instability issue [41].

Based on the PBH criterion, a system is detectable if and only if the PBH matrix (given below) has full column rank for all the eigenvalues, s , or the undetectable modes have an eigenvalue with a negative real part, i.e., are stable [41]:

$$PBH = \begin{bmatrix} sI - A_{ac} \\ H_{ac} \end{bmatrix} \tag{31}$$

In the above equation, to test the detectability, it suffices to check the rank of matrix for the eigenvalues of the augmented system which consist of eigenvalues of the dynamic system and zeros accounted for the unknown forces [31]. Expanding PBH using (17) and (19) it would be easier to investigate stability for different measurements:

$$PBH = \begin{bmatrix} sI & -I & 0 \\ M^{-1}K & sI + M^{-1}C & -M^{-1}S_f \\ 0 & 0 & sI \\ S_s + S_d - S_a M^{-1}K & S_v - S_a M^{-1}C & S_a M^{-1}S_f \end{bmatrix} \tag{32}$$

In the case when acceleration only is measured in all DOFs, $S_s = S_d = S_v = 0$ and $S_a = I$, then in the PBH matrix, the first and last columns are linearly dependent at $s = 0$, as shown in Equation (33). Therefore, the system is not observable when acceleration measurements only are used. The same problem occurs, when velocity only is measured:

$$PBH = \begin{bmatrix} 0 & -I & 0 \\ M^{-1}K & M^{-1}C & -M^{-1}S_f \\ 0 & 0 & 0 \\ -M^{-1}K & -M^{-1}C & M^{-1}S_f \end{bmatrix} \tag{33}$$

Another reason to consider zero eigenvalue, i.e., $s = 0$, to check the detectability is to account for possible uncertainties available in the systems, numerical modeling, and response measurements. For real dynamic systems, the case having zero eigenvalue do not exist. But such uncertainties can result in zero eigenvalue in the observation. Thus, detectability of the system should be ensured for all eigenvalues, including zero eigenvalue.

In the case of displacement measurements available, regardless of whether full or partial displacement measurements are used, the system is detectable in the presence of system damping [41]. However, measuring absolute displacements in the field is a challenging task for full-scale civil structures.

For the case where strain measurements are used, considering the observation matrix which contains the strain selection matrix, see Equation (30), the same argument made for displacement measurements hold and the system would be detectable in the presence of damping even for partial strain measurements. When accelerations together with strains are measured, i.e., the Multi-metric observation, $S_s \neq 0, S_d = S_v = 0$ and $S_a \neq 0$; then the PBH matrix becomes:

$$PBH = \begin{bmatrix} sI & -I & 0 \\ M^{-1}K & sI + M^{-1}C & -M^{-1}S_f \\ 0 & 0 & sI \\ S_s - S_a M^{-1}K & -S_a M^{-1}C & S_a M^{-1}S_f \end{bmatrix} \tag{34}$$

Even if accelerations and strains for all DOFs are not measured, in presence of damping the non-observable modes would be stable and therefore detectability of the system is satisfied. Instead of using acceleration and strain measurements, one potential solution may be using acceleration measurements to estimate velocity and displacement via numerical integration. This may seem a straightforward and easy task but indeed it is not. Especially it is very difficult to handle nonzero direct current (DC) value or low frequency components even in numerical simulation (see pages 51–55 in [51]). In practice, accelerometers are not sensitive to very low frequency and they can't measure static forces. Moreover, data acquisition (DAQ) systems have some drift and it is complicated to separate such effects from low frequency forces. Therefore, integrating acceleration numerically does not provide us with the correct estimate of velocity and displacement.

2.5. Strain Selection Matrix for Planar Truss

Structural deformations of many civil structures can be reasonable assumed to be small, compared with the size of the structure, during their operation. Exploiting this small deflection assumption, it is possible to linearly relate the strains to the displacements (states) with high accuracy. To obtain the linear relation between strains and displacements in a planar truss structure, a general deflection of part of the truss is shown in Figure 2 (left). To see how the strains can be estimated from displacements using a linear relation, it would be simpler if the deformed shape is translated such that one of the shifted nodes coincide with its initial position, that is shifting i' to i (right in Figure 2).

Considering the geometrical relation, when the deformation is small, using Equation (35) strain is estimated from the states linearly:

$$\Delta \approx (D_{2j-1} - D_{2i-1}) \cos(\theta) + (D_{2j} - D_{2i}) \sin(\theta) \tag{35}$$

In matrix form:

$$\epsilon = \frac{\Delta}{L} = \begin{bmatrix} 0 & \dots & \frac{-\cos(\theta)}{L} & \frac{-\sin(\theta)}{L} & \dots & \frac{+\cos(\theta)}{L} & \frac{+\sin(\theta)}{L} & \dots & 0 \end{bmatrix} \begin{Bmatrix} D_1 \\ \vdots \\ D_{2i-1} \\ D_{2i} \\ \vdots \\ D_{2j-1} \\ D_{2j} \\ \vdots \\ D_{2N} \end{Bmatrix} = S_s u(t) \tag{36}$$

To better understand the selection matrices S_a , S_v , S_d , S_s , and S_f consider the simple one-dimensional lumped mass 4-DOF system shown in Figure 3:

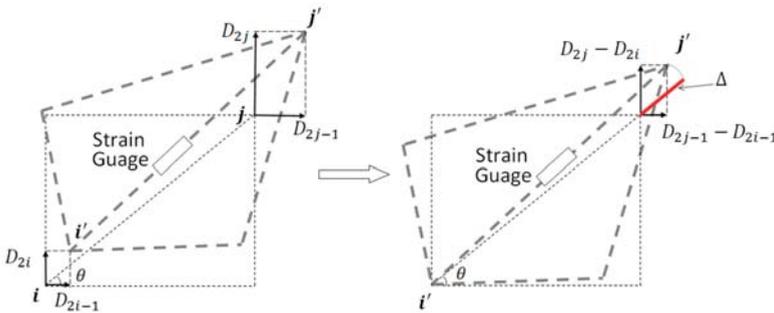


Figure 2. General exaggerated deformation of part of the truss structure (Left)—translated deformed shape to coincide the node “i” with its initial undeformed shape (Right).

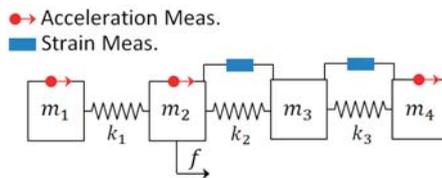


Figure 3. Four Degree of Freedom system with one excitation on the 2nd DOF.

A force (f) is applied on the 2nd mass (DOF) and the acceleration of the 1st, 2nd, and 4th DOF are measured, and the strain between the 2nd and 3rd and between 3rd and 4th DOF (masses) are measured. In the observation (measurement) vector, the first row has information of acceleration of first mass; second row has information of acceleration of second mass; third row has information of acceleration of fourth mass; fourth row has the strain between 2nd and 3rd DOF and the fourth row has the information of the strain between 3rd and 4th DOF:

$$S_a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, S_d = S_v = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, S_s = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, S_p = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

2.6. Strain Selection Matrix for Planar Truss

In Finite Element Methods for solid mechanics, strain and nodal point displacements are related via the following equation:

$$\{\epsilon\} = TN\{u\} = S_s u(t) \tag{37}$$

The product of differentiation matrix operator *T*, and shape function matrix *N* is called displacement differentiation matrix. The differentiation matrix operator for different solid mechanics elements such as beams, plates and shells can be found in solid mechanics finite element methods books. For example see Chapters 2 and 3 in [52].

3. Simulations and Results

Responses of a planar steel truss structure model (see Figure 4) with 18 joints and 34 members are simulated using the fourth order Runge-Kuta method [53]. The time step for response simulation is 1/4096 s. Quality of the simulation is validated by calculating the multiple coherence function [54] shown in Figure 5. The coherence function is a scalar value between zero and one; in which one indicates the outputs are fully due to the inputs. At resonance, there are some drops; which are partly due to small numerical errors in solving the equation of motion and mainly due to spectral leakage and biased and random errors of spectral estimations (see [51] pp. 205–240), because finite amount of data is used to do so. The structure with its node and element numbers is depicted in Figure 4, and the properties of the elements are given in Table 1. A modal damping ratio of 0.7% was considered for all the modes of the structure. In Figure 4, red bold arrows show the applied force locations in the structure; four different excitations are simultaneously applied to the system. H16,16—Frequency Response Function (FRF) of the system between measured DOF “16” and excitation at DOF “16”, called point FRF—is shown with a dotted line in Figure 6. DOF “16” is vertical direction of joint 8. This system is used for response simulations. To have realistic situation for force estimation, the same system used for simulation is not used in the Kalman Filter. This is done because a perfect model of the structure almost never happens. A slightly different model which is 5% stiffer is used representing a modeling error (5%). The point FRF (H16,16) of this stiffer model used in K-F is shown by red solid line in Figure 6.

Several simulations are conducted with different combinations of responses, such as accelerations, displacements, and strains for estimation of the applied forces using the augmented Kalman filter method. For simulating the strain, *S_{ij}(t)*, which is the strain of the link with the end nodes “*i*” and “*j*”, the following formula is used:

$$S_{ij}(t) = \frac{\sqrt{[(X_i + D_{2i-1}) - (X_j + D_{2j-1})]^2 + [(Y_i + D_{2i}) - (Y_j + D_{2j})]^2} - \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}$$

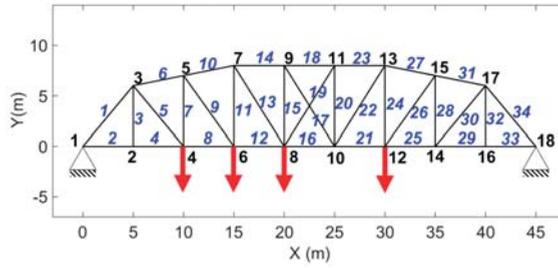


Figure 4. The model of the truss used for simulation—Blue italic numbers show the member number and the black ones indicate the node numbers. There are eighteen nodes and 34 members; Red bold arrows show where the simultaneous excitations are applied.

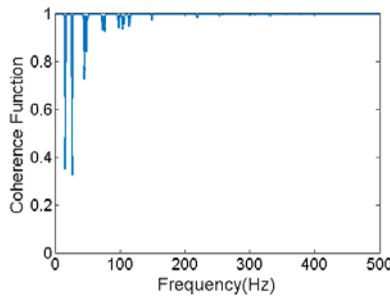


Figure 5. Multiple coherence function is always between zero and one. One indicates all the measured output is due to the considered inputs while zero means other factors (noises, unconsidered input, etc.) have caused the output(s).

Table 1. Properties of the truss structure.

Elements	Young Modulus (Pa)	Cross Section Area (cm ²)	Density Kg/m ³
1, 6, 10, 14, 18, 23, 27, 31, 34	200 × 10 ⁹	15	7800
2, 3, 4, 7, 8, 11, 12, 15, 16, 20, 21, 24, 25, 28, 29, 32, 33	200 × 10 ⁹	9.75	7800
5, 9, 13, 17, 19, 22, 26, 30	200 × 10 ⁹	4.75	7800

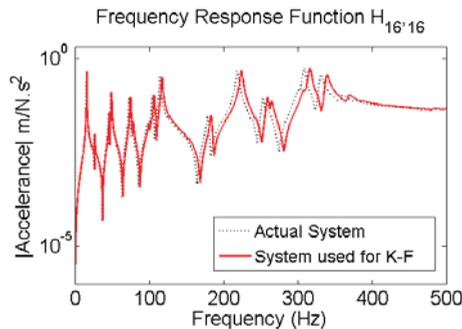


Figure 6. Dotted black: FRF of the system that is used for simulation of response; Solid red line: FRF of the perturbed system, to represent modeling error (5%) used for input estimation.

In Figure 7 typical accelerations and strains responses from the numerical simulations, when the system is subjected to the combination of different forces simultaneously applied to the structure, are shown. In the right of Figure 7, low-frequency variations can be clearly observed in the strain data due to very low frequency excitations applied to the system, but not in acceleration data; which are intrinsic characteristics of acceleration and strain measurements.

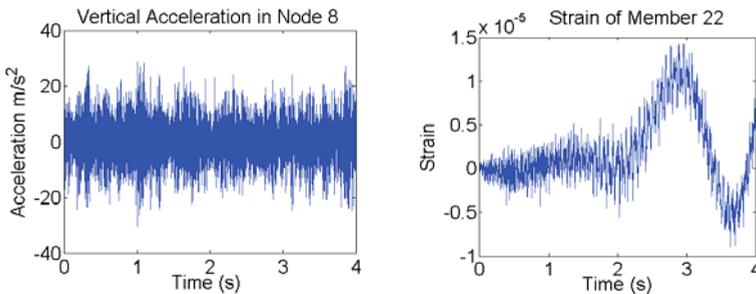


Figure 7. Simulated acceleration (**Left**) in vertical direction at Node “8” and strain (**Right**) in the member “22” when the forces shown in Figure 11 (dotted black) are applied to the structure simultaneously. Low frequencies (almost DC components) are seen in the strain measurements.

If there are forces with zero mean value and no (or very small) measurement noise or modeling error, it is possible to estimate the inputs using only acceleration. However, if different combinations of loading are applied and some of them have non-zero mean values then acceleration-based estimation is misleading. Different combinations of responses are simulated to use for force estimation:

Case 1

- Only acceleration measurements are used.
- System is subjected to random excitations (vertical direction at joints “4”, “8”, and “12”) and impulse applied in the vertical direction of joint “6”.
- Regarding presence of noise and modeling errors, two cases are considered: i) without modeling error and measurement noise and ii) with modeling error (5%) and measurement noise (2%).

Case 2

- Another case that only acceleration measurements are used.
- But, different forces are simultaneously applied at four nodes in vertical direction. Random (vertical direction of Joint “4”), impulsive (vertical direction of Joint “6”), random + low varying high amplitude (vertical direction of Joint “8”), noise + ramp shape (vertical direction of Joint “12”).
- Whether errors are available or not, two cases are considered: (i) without modeling error and measurement noise and (ii) with them where the results are again unstable as in case 1 when noise was considered.

Case 3

- The case that only strain measurements are used.
- Loading condition is the same as Case 2
- Modeling errors and measurement noises are considered.

Case 4

- The case that both acceleration and strain measurements are used.
- Loading condition is the same as Case 2
- Modeling errors and measurement noises are considered.

3.1. Case 1: Acceleration Measurements Only—Random and Impulsive Excitation

3.1.1. No Modeling and Error, No Measurement Noise

The model shown in Figure 4 is subjected to random excitations and impulse. Uncorrelated random forces are exciting the structure in the vertical direction at nodes 4, 8, and 12 and impulsive force is applied at node 6 also in the vertical direction. The applied forces are shown with dotted black lines in Figure 8. Acceleration responses are simulated and measured in vertical direction at nodes 2, 4, 5, 6, 7, 8, 9, 12, 14, 15 and in horizontal direction at nodes 4, 5, 8, and 9.

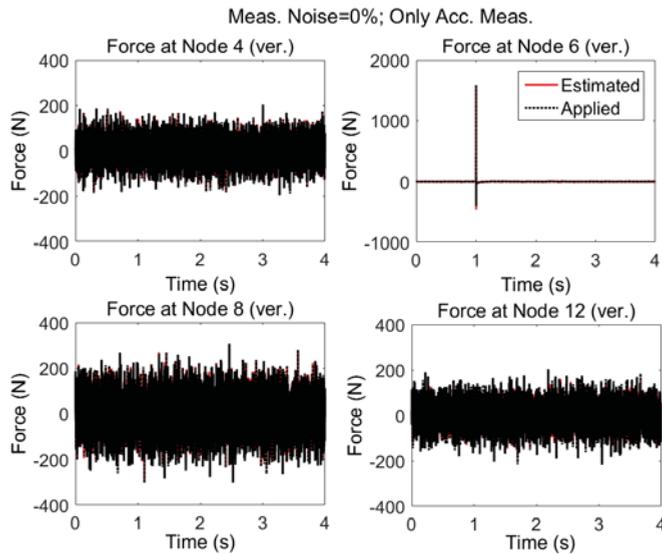


Figure 8. System is subjected to three uncorrelated random noise in vertical directions at nodes 4, 8, and 12 and impulse applied at node 6 (vertical). Excitations are estimated by AKF only using the accelerations; neither modeling nor measurement error is considered.

Using the accelerations and AKF algorithm, without considering measurement noise and modeling errors, the applied forces are estimated. The results are shown in Figure 8. In Figure 9, the estimated loads are shown for a smaller time interval to see the accuracy of estimation.

It is seen that it is possible to estimate the excitation(s) using only acceleration measurements provided that:

- 1) there is no measurement noise and modeling error, and
- 2) the structure is subjected to random zero mean and impulsive forces, Then, even when forces are applied simultaneously at different locations of the structure, it is possible to estimate the input loading reliably. However, assuming no measurement errors and no discrepancy between the dynamic response of model and the real structure is too far from reality.

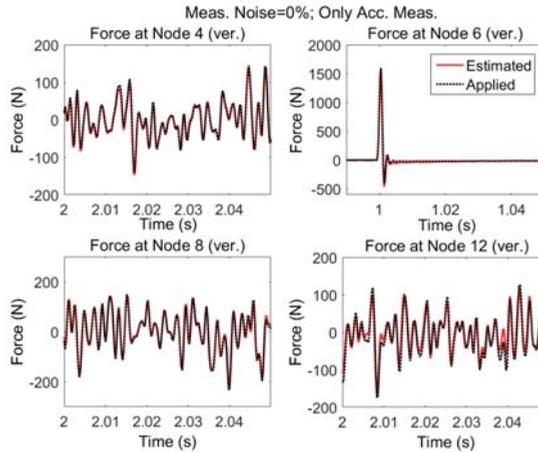


Figure 9. Estimated and applied forces shown for smaller time interval: detail of Figure 8.

3.1.2. Modeling Error (5%) and Measurement Noise (2%)

In practice it is not possible to have an exact model, nor is it possible to have noise free measurements, so we have considered such uncertainties for force estimation. The results are shown in Figure 10. It is seen that the estimation becomes unstable; which is attributed to the fact that the AKF to use only acceleration data in the measurement update stage is not detectable as described in the Section 2.4. For undetectable systems, the states cannot be reliably estimated with the observed information. Such instability issues have been reported in other papers as well [40,41].

Therefore, when the structure is subjected to multiple zero mean random and impulsive forces, only acceleration measurements can be misleading to estimate the applied forces if there are measurement noises or modeling errors. Since such uncertainties are inevitable in almost all real-world situations, we can conclude that only acceleration measurements will end in erroneous force approximation.

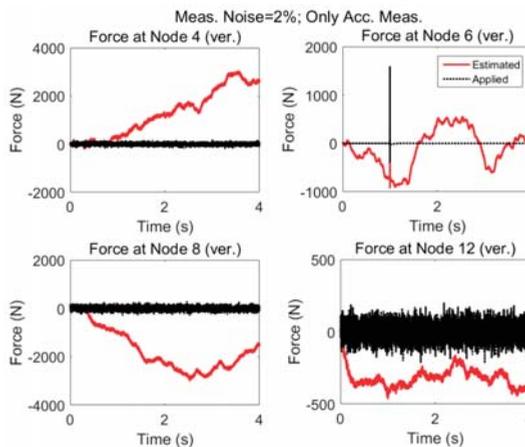


Figure 10. Same loading as in Figure 8, but in this case both the measurement noises (2%) and modelling errors (5%) are present.

3.2. Case 2: Acceleration Measurements Only—Presence of Low Varying and Non-Zero Mean Excitations

3.2.1. No Modeling and Error, No Measurement Noise

The measurement configuration is same as “case 1” but the excitations are different statistically; that is in addition to random and impulsive load applied in vertical direction of nodes 4 and 6, respectively, low varying plus random excitation and ramp shape plus random excitation are applied to the nodes 8 and 12. The applied forces are shown with dotted black lines and estimated forces by the AKF are shown with red solid lines in Figure 11. Even without considering any additional uncertainties, such as measurement noise and modelling error, it is seen that only acceleration measurements are not effective in estimating low-frequency (at node 8) and nonzero-mean excitation (at node 12); also, zero mean random (at node 4) and impulsive load (at node 6). Estimations by AKF are highly biased in the presence of nonzero low varying forces. This instability in the presence of nonzero mean low varying excitations can be attributed to the rigid body modes as was discussed in the case of eigenvalues equal to zero in Section 2.4.

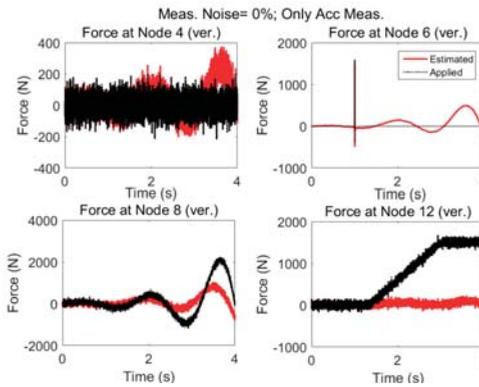


Figure 11. System is subjected simultaneously to four different forces where some of them have nonzero mean and low frequency and high amplitude. No error (modeling and measurement error) is added and excitation is estimated by AKF only using the accelerations. Even without added uncertainties relying solely on acceleration will fail to estimate the inputs.

Power spectrums of estimated and applied forces at two locations are given in Figure 12. It is seen, even though no modeling error or measurement noises are considered, using acceleration only the low frequency components cannot be captured.

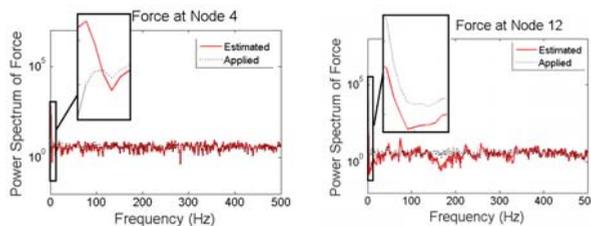


Figure 12. Frequency contents of applied and estimated forces at nodes 4 and 12 for “case 2”.

When there are nonzero low varying forces applied to the structure, it is not possible to obtain good estimates of applied forces based on only acceleration measurements even if there is no measurement noise nor modeling error. Specifically, low frequencies cannot be captured in this case.

3.2.2. Modeling Error (5%) and Measurement Noise (2%)

The results for the acceleration only case when modeling error and measurement noises are present are shown in Figure 13. As expected, the estimation fails to approximate the excitations.

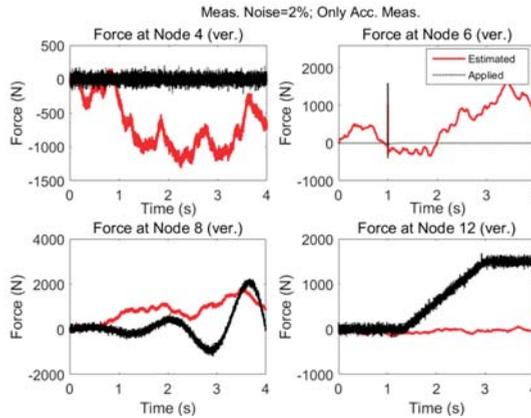


Figure 13. Same as Case 2 but there is modeling error (5%) and measurement noise (2%).

As expected, in the presence of nonzero low varying forces and modeling errors and measurement noises there will be large deviations in the estimated excitations.

3.3. Case 3: Only Strains Are Measured—Modeling Error (5%) and Measurement Noise (2%)

Only strain measurements would be attractive to consider; for this case of force reconstruction based only on strain measurements, Strain of members 4, 5, 9, 12, 13, 16, 21, 24, and 26 are simulated. The results of estimated forces are shown in Figure 14.

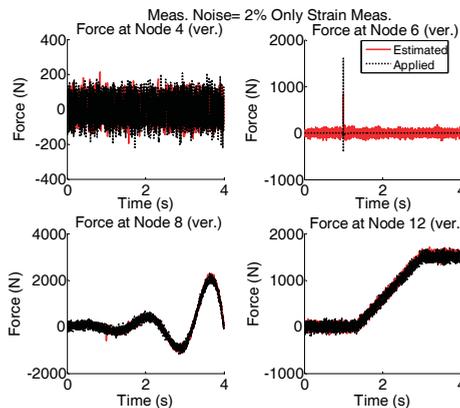


Figure 14. System is simultaneously subjected to four different forces shown with dotted black; Modeling (5%) and measurement error (2% of rms of the random signal) is added and excitation is estimated by AKF using the strain measurements with single acceleration measurement.

Again, the results shown in Figure 14 are plotted for shorter time interval and are given in Figure 15.

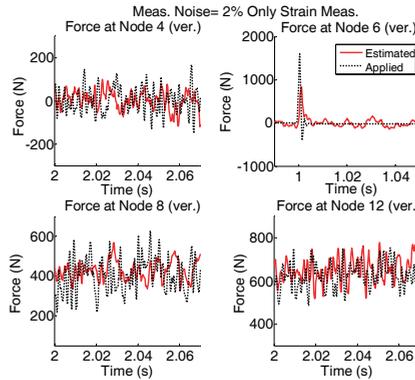


Figure 15. The results shown in Figure 14 are re-plotted for shorter time intervals to see the quality of estimation.

In the case of only strain measurements, it is possible to capture the low variations of the force even when both measurement noises and modeling errors are present. However, in high frequencies the variations are not captured reasonably.

3.4. Case 4: Strain and Acceleration—Modeling Error (5%) and Measurement Noise (2%)

Since strain measurements are simpler and more accessible than displacement measurements, a combination of acceleration and strain measurements for force estimation using AKF are investigated. The linear estimation of strains based on displacements is given in Equation (35) for planar truss structure.

The applied forces to the system are same as “case 2”. Modeling error and measurement noises are considered. The accelerations are measured in vertical directions at nodes 4, 6, 8, and 12 and in the horizontal direction at node 6; Strains are measured in members 5, 9, 13, and 26. To see the quality of the estimated forces, the results shown in Figure 16 are plotted for shorter time interval and are given in Figure 17.

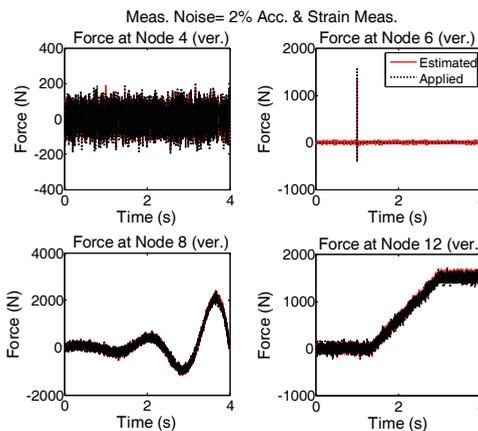


Figure 16. System is simultaneously subjected to four different forces shown with dotted black; Modeling and measurement error (2% of rms of the random signal) is added and excitation is estimated by AKF using the accelerations and strain measurements.

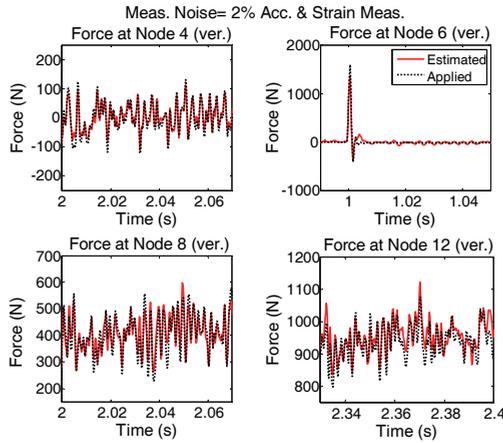


Figure 17. The results shown in Figure 16 are re-plotted for shorter time intervals to see how the estimation follows the applied forces in more detail.

It is seen that in the multi-metric case, i.e., deploying strain and acceleration measurements together, the best results are achieved in low-frequency as well as high-frequency and the estimation is stable. Frequency content of estimated and applied forces at nodes 4 and 12 are shown in Figure 18.

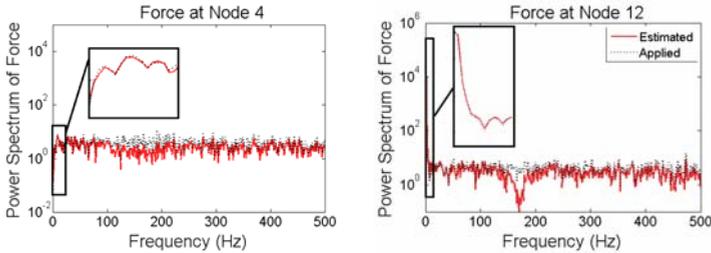


Figure 18. Frequency content of applied and estimated forces at Nodes 4 and 12 for “case 3”.

The results of the above-mentioned cases are summarized in Table 2. For the sake of conciseness, the cases when there is no measurement noise and modelling errors were not discussed for only strain measurements (case 3) and multi-metric measurements (case 4).

Table 2. Results summary for different cases. The symbols are: Yes (✓), No (×), and Not Applicable (-).

Configurations and Frequency Ranges	Random and Impulsive Excitation		Random and Impulsive Excitations + Low Varying and Non-Zero Mean Excitations	
	No Measurement Noise & No Modelling Error	2% Measurement Noise & 5% Modelling Error	No Measurement Noise & No Modelling Error	2% Measurement Noise & 5% Modelling Error
Acc.	Low Frequency	-	×	×
	High Frequency	✓	×	×
strain	Low Frequency	-	✓	✓
	High Frequency	×	×	×
Acc. + strain	Low Frequency	-	✓	✓
	High Frequency	✓	✓	✓

3.5. Comparison of Different Types of Measurements

For the purpose of comparing the quality of input estimation based on different types of responses or their combinations, the root mean square of the difference between estimated and applied force vectors is calculated using Equation (38) and is plotted in Figure 19:

$$RMS\ error = \sqrt{\frac{1}{N_i} \sum_{i=1}^{N_i} (F(i) - \hat{F}(i))^2} \tag{38}$$

in the above formula, N_i is number of samples in the force vector, F is the applied (reference) force and \hat{F} is the estimated force.

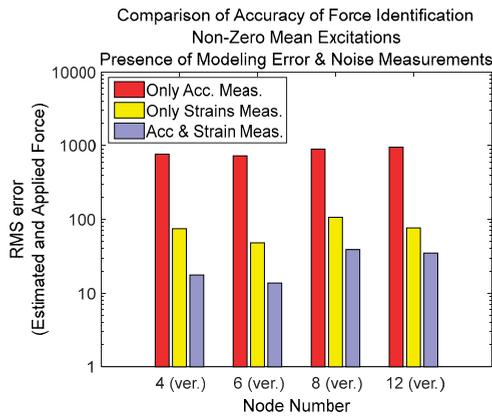


Figure 19. RMS error between the estimated and applied forces for different types of measurements.

NOTE: It is also possible to use other measures to check the quality of input estimation; for example, one possible way is to compare norm of the estimated vector to the norm of reference vector divided by the norm of reference vector:

$$error = 100 \times \frac{\|F\| - \|\hat{F}\|}{\|\hat{F}\|} \tag{39}$$

but in this case if low frequency estimation or the trend of estimation has good accuracy, then by increasing the amplitude of low varying excitation the RMS error will be reduced. In this case, the quality of estimation depends on the magnitude low frequency components and is misleading. Therefore, we have selected former formula in order to check the quality of estimations.

It is seen that in the case of multiple loads simultaneously applied to the system where some of them have low varying nonzero mean components, then if only accelerations are measured then the error of estimation is very high compared to the case of only strain measurements. The best result is obtained when multi-metric measurements, strain and acceleration, is used.

3.6. Effect of Different Configurations for Strain and Acceleration Measurements

Four different strain and acceleration arrangements to measure the responses of the structure are considered (Figure 20). The quality of input force estimations for the applied forces described in “case 2”, when modeling errors and measurement noises are present, are compared and plotted in Figure 21.

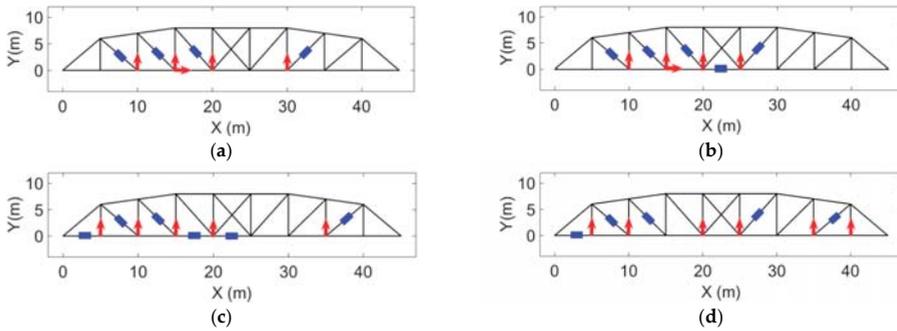


Figure 20. Four different strain and acceleration measurement configurations; Acceleration measurement shown by red arrow and strain measurement shown by blue rectangle. (a) Set 1; (b) Set 2; (c) Set 3; (d) Set 4.

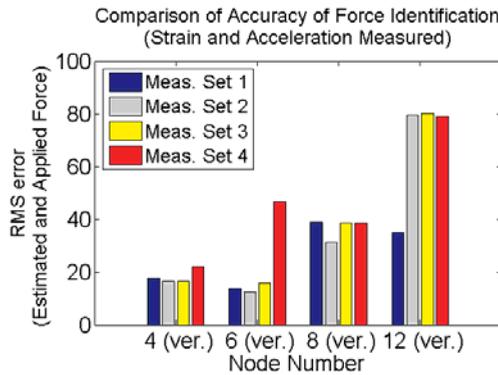


Figure 21. RMS error of estimated and applied forces for different strain gauges and accelerometer configurations on the truss structure, in presence of modeling error and measurement noises. Applied forces are dotted black line in Figure 11.

Measurement configuration (a) was used for force estimation and the results are shown in case 4. Using a different measurement configuration (b) shown in Figure 20, the same input forces applied to the structure in case 4 are estimated for the purpose of Multi-metric measurement configuration and the results are shown in Figure 22 and for smaller time interval in Figure 23.

It is seen that the results in the former sensor configuration (set "a") for the strain and acceleration is better particularly the estimated force at Node 12. Generally, having sensors closer to the input locations increases the quality and accuracy of estimation [18,41]. It is observed that for the current formulation, using multi-metric measurements solves the stability issue. But if accurate estimation of excitation for a certain location is desired, there should be a measurement point close to that location.

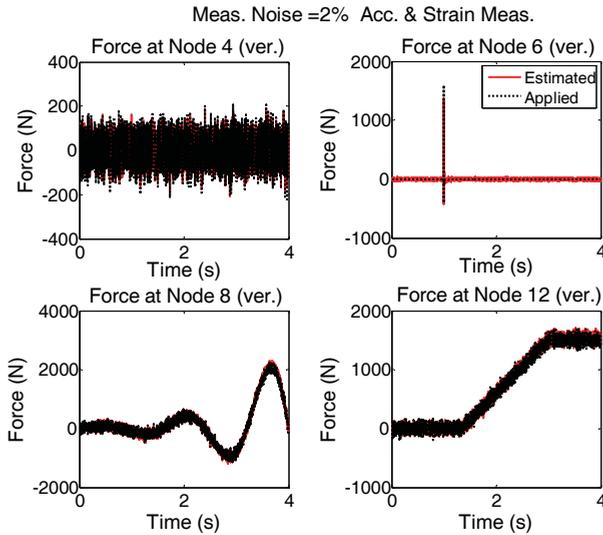


Figure 22. Same as Figure 16 but the measurement configuration is different. Sensor arrangement is shown in Figure 20b.

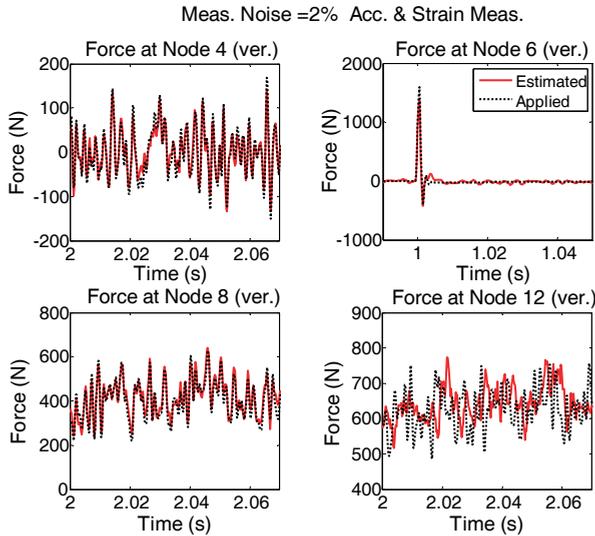


Figure 23. The results shown in Figure 22 are re-plotted for shorter time intervals to see the quality of estimation.

4. Discussion and Conclusions

In this study, incorporation of strain measurements together with acceleration measurements in AKF process has been explored to provide a reliable and accurate force identification scheme for dynamic structures. Incorporating Multi-metric observations in AKF has been formulated and the method has been numerically validated using a planar truss structure simultaneously subjected to various types of forces. The proposed Multi-metric AKF approach showed significant improvements

both in reliability and accuracy of dynamic force estimations under uncertain numerical modeling and measuring environment.

When only acceleration measurements, which are cheap and easy-to-obtain in practice, were used, AKF was found to be reliable just for the cases where none of the measurement noises, modeling errors, low frequency and nonzero forces are present (see Table 2). Use of displacement measurements resolves the stability issues, but are expensive and difficult to deploy in practice. On the other hand, strain measurements are easier and more practical in many cases. As a remedy to the inaccuracy and instability with the force estimation problem, AKF to use both acceleration and strain measurements was formulated. And its efficacy was validated using numerical simulation of a truss structure subjected to broad band excitations. To have more realistic simulation, measurement noises (2%) and modeling error (5%) were considered in AKF. additionally, it was observed that using only strain measurements for AKF can estimate the low-varying components of excitation with good accuracy but not for the high-frequency components. The results for different measurement configurations and for different cases were summarized in Table 2. In the future, researching the optimization of measurement configuration will be carried out and experiments on the proposed multi-metric approach for AKF will be conducted both in lab- and field-scale testbeds to further examine its performance.

Author Contributions: B.K. led the numerical validation and writing the paper. D.M. helped numerical modeling and reviewed the paper. H.J. helped AKF theory formulation and advised whole research process.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, J.J.; Ma, C.K.; Kung, I.C.; Lin, D.C. Input force estimation of a cantilever plate by using a system identification technique. *Comput. Methods Appl. Mech. Eng.* **2000**, *190*, 1309–1322. [CrossRef]
2. Ma, C.K.; Lin, D.C. Input forces estimation of a cantilever beam. *Inverse Probl. Eng.* **2000**, *8*, 511–528. [CrossRef]
3. Harvey, D.; Cross, E.; Silva, R.A.; Farrar, C.; Bement, M. Input Estimation from Measured Structural Response. In *Structural Dynamics*; Springer: New York, NY, USA, 2011; Volume 3, pp. 689–708.
4. Guillaume, P.; Parloo, E.; Verboven, P.; De Sitter, G. An Inverse Method for the Identification of Localized Excitation Sources. In Proceedings of the 20th International Modal Analysis Conference (IMAC XX), Los Angeles, CA, USA, 4–7 February 2002.
5. Bateman, V.I.; Mayes, R.L.; Carne, T.G. Comparison of Force Reconstruction Methods for a Lumped Mass Beam. *Shock Vib.* **1997**, *4*, 231–239. [CrossRef]
6. Ma, C.K.; Tuan, P.C.; Lin, D.C.; Liu, C.S. A study of an inverse method for the estimation of impulsive loads. *Int. J. Syst. Sci.* **1998**, *29*, 663–672. [CrossRef]
7. Hollandsworth, P.E.; Busby, H.R. Impact force identification using the general inverse technique. *Int. J. Impact Eng.* **1989**, *8*, 315–322. [CrossRef]
8. Nordström, L.J.; Nordberg, T.P. A time delay method to solve non-collocated input estimation problems. *Mech. Syst. Signal Process.* **2004**, *18*, 1469–1483. [CrossRef]
9. Nordberg, T.P.; Gustafsson, I. Using QR factorization and SVD to solve input estimation problems in structural dynamics. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 5891–5908. [CrossRef]
10. Nordstrom, L.J.; Nordberg, T.P. A critical comparison of time domain load identification methods. In Proceedings of the 6th International Conference on Motion and Vibration Control, Saitama, Japan, 20–24 August 2002.
11. Wang, B.T. Prediction of impact and harmonic forces acting on arbitrary structures: Theoretical formulation. *Mech. Syst. Signal Process.* **2002**, *16*, 935–953. [CrossRef]
12. Muszynska, A. Vibrational diagnostics of rotating machinery malfunctions. *Int. J. Rotat. Mach.* **1995**, *1*, 237–266. [CrossRef]
13. Olsson, M. On the fundamental moving load problem. *J. Sound. Vib.* **1991**, *145*, 299–307. [CrossRef]
14. Zhu, X.Q.; Law, S.S. Moving load identification on multi-span continuous bridges with elastic bearings. *Mech. Syst. Signal Process.* **2006**, *20*, 1759–1782. [CrossRef]

15. Kreitinger, T.J. *Force Identification from Structural Response (No. WL-TR-89-81)*; Weapons Lab (AFSC) Kirtland AFB NM: Albuquerque, NM, USA, 1990.
16. Bateman, V.I.; Carne, T.G.; Mayes, R.L. Force reconstruction using a sum of weighted accelerations technique. In Proceedings of the 10th International Modal Analysis Conference, San Diego, CA, USA, 3–7 February 1992.
17. Sharif-Khodaei, Z.; Ghajari, M.; Aliabadi, M.H. Determination of impact location on composite stiffened panels. *Smart Mater. Struct.* **2012**, *21*. [CrossRef]
18. Ghajari, M.; Sharif-Khodaei, Z.; Aliabadi, M.H.; Apicella, A. Identification of impact force for smart composite stiffened panels. *Smart Mater. Struct.* **2013**, *22*. [CrossRef]
19. Sun, H.; Büyükoztürk, O. Identification of traffic-induced nodal excitations of truss bridges through heterogeneous data fusion. *Smart Mater. Struct.* **2015**, *22*. [CrossRef]
20. Mao, Y.M.; Guo, X.L.; Zhao, Y. A state space force identification method based on Markov parameters precise computation and regularization technique. *J. Sound Vib.* **2010**, *329*, 3008–3019. [CrossRef]
21. Li, Z.; Feng, Z.P.; Chu, F.L. A load identification method based on wavelet multi-resolution analysis. *J. Sound Vib.* **2014**, *333*, 381–391. [CrossRef]
22. Ding, Y.; Law, S.S.; Wu, B.; Xu, G.S.; Lin, Q.; Jiang, H.B.; Miao, Q.S. Average acceleration discrete algorithm for force identification in state space. *Eng. Struct.* **2013**, *56*, 1880–1892. [CrossRef]
23. Bernal, D.; Ussia, A. Sequential deconvolution input reconstruction. *Mech. Syst. Signal Process.* **2015**, *50–51*, 41–55. [CrossRef]
24. Lai, T.; Yi, T.-H.; Li, H.-N. Parametric study on sequential deconvolution for force identification. *J. Sound Vib.* **2016**, *377*, 76–89. [CrossRef]
25. Ma, C.K.; Chang, J.M.; Lin, D.C. Input forces estimation of beam structures by an inverse method. *J. Sound Vib.* **2003**, *259*, 387–407. [CrossRef]
26. Tuan, P.C.; Ji, C.C.; Fong, L.W.; Huang, W.T. An input estimation approach to on-line two-dimensional inverse heat conduction problems. *Numer. Heat Transf. Fundam.* **1996**, *29*, 345–363. [CrossRef]
27. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; Department of Computer Science University: Chapel Hill, NC, USA, 2000.
28. Hoshiya, M.; Saito, E. Structural identification by extended Kalman filter. *J. Eng. Mech. ASCE* **1984**, *110*, 1757–1770. [CrossRef]
29. Yang, J.N.; Lin, S.; Huang, H.; Zhou, L. An adaptive extended Kalman filter for structural damage identification. *Struct. Control Health Monit.* **2006**, *13*, 849–867. [CrossRef]
30. Ghanem, R.; Ferro, G. Health monitoring for strongly non-linear systems using the Ensemble Kalman filter. *Struct. Control Health Monit.* **2006**, *13*, 245–259. [CrossRef]
31. Kivman, G.A. Sequential parameter estimation for stochastic systems. *Nonlinear Process. Geophys.* **2003**, *10*, 253–259. [CrossRef]
32. Miller, R.N.; Carter, E.F.; Blue, S.T. Data assimilation into nonlinear stochastic models. *Tellus Ser. A Dyn. Meteorol. Oceanogr.* **1999**, *51*, 167–194. [CrossRef]
33. Khalil, M.; Sarkar, A.; Adhikari, S. Nonlinear filters for chaotic oscillatory systems. *Nonlinear Dyn.* **2009**, *55*, 113–137. [CrossRef]
34. He, J.; Huang, Q.; Xu, Y.L. Synthesis of vibration control and health monitoring of building structures under unknown excitation. *Smart Mater. Struct.* **2014**, *23*. [CrossRef]
35. Palanisamy, R.; Cho, S.; Kim, H.; Sim, S.H. Experimental validation of Kalman filter-based strain estimation in structures subjected to non-zero mean input. *Smart Mater. Struct.* **2015**, *15*, 489–503. [CrossRef]
36. Jo, H.; Spencer, B.F., Jr. Multi-scale Model-based Structural Health Monitoring. In Proceedings of the SPIE, San Diego, CA, USA, 9–13 March 2014; Volume 9061.
37. Vester, R.; Percin, M.; van Oudheusden, B. Deconvolution Kalman filtering for force measurements of revolving wings. *Meas. Sci. Technol.* **2016**, *27*. [CrossRef]
38. Wahrburg, A.; Bos, J.; Listmann, K.D.; Dai, F.; Matthias, B.; Ding, H. Motor-Current-Based estimation of cartesian contact forces and torques for robotic manipulators and its application to force control. *IEEE Trans. Autom. Sci. Eng.* **2017**, *1–8*. [CrossRef]
39. Capurso, M.; Ardakani, M.M.G.; Johansson, R.; Robertsson, A.; Rocco, P. Sensorless kinesthetic teaching of robotic manipulators assisted by observer-based force control. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 945–950.

40. Lourens, E.; Reynders, E.; De Roeck, G.; Degrande, G.; Lombaert, G. An augmented Kalman filter for force identification in structural dynamics. *Mech. Syst. Signal Process.* **2012**, *27*, 446–460. [CrossRef]
41. Naets, F.; Cuadrado, J.; Desmet, W. Stable force identification in structural dynamics using Kalman filtering and dummy-measurements. *Mech. Syst. Signal Process.* **2015**, *50*, 235–248. [CrossRef]
42. Lourens, E.; Papadimitriou, C.; Gillijns, S.; Reynders, E.; De Roeck, G.; Lombaert, G. Joint input-response estimation for structural systems based on reduced-order models and vibration data from a limited number of sensors. *Mech. Syst. Signal Process.* **2012**, *29*, 310–327. [CrossRef]
43. Maes, K.; Smyth, A.W.; De Roeck, G.; Lombaert, G. Joint input-state estimation in structural dynamics. *Mech. Syst. Signal Process.* **2016**, *70*, 445–466. [CrossRef]
44. Azam, S.E.; Chatzi, E.; Papadimitriou, C. A dual Kalman filter approach for state estimation via output-only acceleration measurements. *Mech. Syst. Signal Process.* **2015**, *60*, 866–886. [CrossRef]
45. Azam, S.E.; Chatzi, E.; Papadimitriou, C.; Smyth, A. Experimental validation of the Kalman-type filters for online and real-time state and input estimation. *J. Vib. Control* **2017**, *23*, 2494–2519. [CrossRef]
46. Pisoni, A.C.; Santolini, C.; Hauf, D.E.; Dubowsky, S. Displacements in a vibrating body by strain gage measurements. In Proceedings of the International Society for Optical Engineering, Trest Castle, Czech Republic, 18–22 September 1995.
47. Santos, F.; Peeters, B.; Menchicchi, M.; Lau, J.; Gielen, L.; Desmet, W.; Goes, L. Strain-based dynamic measurements and modal testing. In Proceedings of the 32nd International Modal Analysis Conference, Orlando, FL, USA, 3–6 February 2014.
48. Juang, J.N.; Phan, M.Q. *Identification and Control of Mechanical Systems*; Cambridge University Press: Cambridge, UK, 2001; Chapter 7.
49. Sim, S.H.; Spencer, B.F., Jr.; Nagayama, T. Multi-metric sensing for structural damage detection. *J. Eng. Mech.* **2010**, *137*, 22–30. [CrossRef]
50. Park, J.W.; Sim, S.H.; Jung, H.J. Displacement estimation using Multi-metric data fusion. *IEEE/ASME Trans. Mechatron.* **2013**, *18*, 1675–1682. [CrossRef]
51. Brandt, A. *Noise and Vibration Analysis: Signal Analysis and Experimental Procedures*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
52. Liu, G.R.; Quek, S.S. *The Finite Element Method: A Practical Course*; Butterworth-Heinemann: Oxford, UK, 2013.
53. Thorby, D. The Linear Single Degree of Freedom System: Response in the Time Domain. In *Structural Dynamics and Vibration in Practice: An Engineering Handbook*; Butterworth-Heinemann: Oxford, UK, 2008; Chapter 3.
54. Josefsson, A.; Magnevall, M.; Ahlin, K.; Broman, G. Spatial location identification of structural nonlinearities from random data. *Mech. Syst. Signal Process.* **2012**, *27*, 410–418. [CrossRef]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

IMU-Based Gait Recognition Using Convolutional Neural Networks and Multi-Sensor Fusion

Omid Dehzangi, Mojtaba Taherisadr *and Raghvendar ChagalVala

Computer and Information Science Department, University of Michigan-Dearborn, Dearborn, MI 48128, USA; dehzangi@umich.edu (O.D.); rchagal@umich.edu (R.C.)

* Correspondence: mojtabat@umich.edu; Tel.: +1-313-603-9109

Received: 17 September 2017; Accepted: 23 November 2017; Published: 27 November 2017

Abstract: The wide spread usage of wearable sensors such as in smart watches has provided continuous access to valuable user generated data such as human motion that could be used to identify an individual based on his/her motion patterns such as, gait. Several methods have been suggested to extract various heuristic and high-level features from gait motion data to identify discriminative gait signatures and distinguish the target individual from others. However, the manual and hand crafted feature extraction is error prone and subjective. Furthermore, the motion data collected from inertial sensors have complex structure and the detachment between manual feature extraction module and the predictive learning models might limit the generalization capabilities. In this paper, we propose a novel approach for human gait identification using time-frequency (TF) expansion of human gait cycles in order to capture joint 2 dimensional (2D) spectral and temporal patterns of gait cycles. Then, we design a deep convolutional neural network (DCNN) learning to extract discriminative features from the 2D expanded gait cycles and jointly optimize the identification model and the spectro-temporal features in a discriminative fashion. We collect raw motion data from five inertial sensors placed at the chest, lower-back, right hand wrist, right knee, and right ankle of each human subject synchronously in order to investigate the impact of sensor location on the gait identification performance. We then present two methods for early (input level) and late (decision score level) multi-sensor fusion to improve the gait identification generalization performance. We specifically propose the minimum error score fusion (MESF) method that discriminatively learns the linear fusion weights of individual DCNN scores at the decision level by minimizing the error rate on the training data in an iterative manner. 10 subjects participated in this study and hence, the problem is a 10-class identification task. Based on our experimental results, 91% subject identification accuracy was achieved using the best individual IMU and 2DTF-DCNN. We then investigated our proposed early and late sensor fusion approaches, which improved the gait identification accuracy of the system to 93.36% and 97.06%, respectively.

Keywords: gait identification; inertial motion analysis; spectro-temporal representation; deep convolutional neural network; multi-sensor fusion; error minimization

1. Introduction

Gait refers to the manner of stepping or walking of an individual. Human gait analysis research dates to the 1960s [1] when it was used for medical purposes for early diagnosis of various disorders such as neurological disorders such as *Cerebral Palsy*, *Parkinson's* or *Rett syndrome* [2], musculoskeletal disorders such as *spinalstenosis* [3], and disorders caused by aging, affecting large percentage of population [4].

Reliable monitoring of gait characteristics over time was shown to be helpful in early diagnosis of diseases and their complexities. More recently, gait analysis has been employed to identify an individual from others. Unlike iris, face, fingerprint, palm veins, or other biometric identifiers,

gait pattern can be collected at a distance unobtrusively [5]. In addition, recent medical studies illustrated that there are 24 various components to human gait and that gait can be unique if all movements are considered [6]. As a result, gait has the potential to be used for biometric identification. It is particularly significant as gait patterns are naturally generated and can be seamlessly used for authentication under smart and connected platforms e.g., keyless smart vehicle/home entry [7], health monitoring [8], etc.

In recent years, there has been much effort on employing wearable devices for activity recognition [9], activity level estimation [10], joint angle estimation [11], activity-based prompting [12], and sports training [13]. Recently, gait identification using wearable motion sensors has become an active research topic because of the widespread installation of sensors for measuring movements in smartphones, fitness trackers, and smartwatches [7,14]. Most of the wearable motion sensors use Micro Electro Mechanical Systems (MEMS) based inertial sensors. These inertial sensors (accelerometers, gyroscopes) are one of the most important members of MEMS family and are combined together as inertial measurement units (IMU). Most modern accelerometers are electromechanical devices that measure acceleration forces in one, two, or three orthogonal axes. Gyroscope sensors are devices that measure angular velocity in three directions. Due to their small-size, portability, and high processing power, IMUs are widely used for complex motion analysis. Hence, gait recognition using wearable IMUs has become an efficient Privacy Enhancing Technology (PET) [15].

Advent of MEMS-based accelerometers and gyroscopes and wireless interfaces such as Bluetooth and Wi-Fi have made the measurement setup for gait analysis data collection non-intrusive and ubiquitous. IMUs have become a significant part of ubiquitous smart devices and therefore integration of inertial sensors in smart devices has become a common practice. There is a mass of people using smart devices on a daily basis. With the latest achievements in the field of pervasive computing, limitations of inertial sensors such as cost, storage, and computational power were overcome to a great extent [16]. Therefore, inertial sensors are not only restricted to simple tasks such as tilt estimation but also for complex tasks such as advanced motion analysis, and activity recognition [14]. They have also been evaluated in medical applications, such as analysis of patient's health based on gait abnormalities [17], fall detection [18]. Although the IMU-based wearables have enabled pervasive motion and gait analysis, there are some intrinsic challenges with those devices. Since, the wearable device is always worn casually, relative orientation between the sensors and the subject body cannot be fixed over different sessions of data acquisition [19]. As the coordinate system used by sensors is defined relative to the frame of the device, small orientation changes of sensor installation may make measurements quite different [19]. The issue of ensuring orientation invariance in extracting gait features has been a matter of concern in many studies. In [20], the authors introduced an orientation invariant measure to alleviate the orientation dependency issue. In this work, we also employ an orientation invariant resulting measures of motion as the input signals.

A large number of research studies have been conducted on developing gait recognition systems using inertial sensors. As a pioneer study in this field, in [21], a triaxial accelerometer was used and fixed on a belt to keep the relative position between the subject's body and sensors unchanged. In order to detect the gait cycles, they have applied a peak detection method. Then, they implemented a template matching process to identify their subjects from their gaits. Due to the less contribution of axis *Y*, authors just used *X* and *Z* axes. Similarity-based measures and machine learning are two frequently used approaches for gait identification in the recent literature. Similarity-based measures such as Tanimoto distance [22], dynamic time warping metrics (DTW) [23], and Euclidean distance [24] have been used in the recent studies. Similarity-based approaches are dependent on selecting representative gait patterns (commonly by an expert and manually) and requires storing them for all subjects in order to compare them with the search population, which, in turn, results in lower efficiency in storage and computation. Machine learning techniques are commonly designed in two major modules after pre-processing of the motion data: (1) feature extraction from the input signal in short windows of the streaming data; and (2) model training to generate a predictive model fed by the data at the feature

space. Various modeling algorithms such as Bayesian network classifier [25], hidden Markov model classifier [24], support vector machines, and decision trees [25] have been used in gait recognition applications. The performance of such systems are highly dependent on the extracted features and their resulting hypothesis class, which is the set of possible predictors with the fixed set of features. The impact of noise interference and particularly motion artifacts on complex sensor data makes the task of extracting relevant and robust features very challenging. Commonly, feature extraction is undertaken manually and via handcraft effort for a specific application [15]. However, extracting manual and hand crafted features for machine learning based systems is cumbersome, subjective, and is prone to biases due to the complexity of sensor data collected from IMUs [26,27]. Manual feature extraction is heuristic and can result in poor expressivity of the feature set (i.e., the set of possible predictors with a fixed set of features may not be good enough). Therefore, the best model given the manual features might generate poor accuracy compared to an optimal performance given the desired representative feature subspace. Another important reason for poor expressivity of commonly used machine learning-based methods can be the detachment of feature extraction and the predictive model training. In this way, important information that might be crucial for high performance predictive modeling, can be neglected in the process of feature extraction.

In this paper, we propose a gait recognition framework and investigate the ability to extract time-invariant signature motion patterns within a gait cycle of each subject to identify him/her from others. We first exploit the information provided by expanding the motion signals recorded from various IMUs worn by the participants to 2D spectro-temporal space via time-frequency (TF) analysis. Due to the non-stationarity of the motion signals, TF and instantaneous frequency (IF)-based methods accommodate the temporal variations in the gait patterns during a gait cycle segment. However, there are two major issues in order to extract relevant descriptors from the 2D TFs: (a) efficient selection of relevant features from the 2D spectro-temporal expanded space might not be feasible due to high dimensionality of the space; and (b) selection of a reliable predictive model given the 2D TFs is a difficult task due to the high dimensionality of the input space and shallow models might face the challenge of curse of dimensionality as discussed in [28]. The authors in [28] discussed that placing decision hyperplanes directly the high dimension space space (the high resolution 2D TF in this work) might raise the risk of curse of dimensionality and hurt the generalization ability of the learnt predictive model. They illustrated that incorporating hierarchical locality using deep learning structures can be sufficient to avoid the curse of dimensionality [28]. Therefore, in this paper, we design a deep convolutional neural network (DCNN) model that is trained for each of the sensor nodes (i.e., five inertial sensors) and the modalities (i.e., accelerometer and Gyroscope readings) to extract *individual signature patterns* from the 2D expanded gait cycles and optimize the *predictive identification model* at the same time in a discriminative fashion. The best individual DCNN performance reaches 91% identification accuracy. In order to aggregate the complementary discriminative information from all sensors, we then investigate multi-sensor early and late fusion with the aim of improving the gait identification performance. We achieve the average accuracy of 93.36% via early fusion by augmenting multi-sensor gait cycles at the input level. In late fusion, a discriminative performance measure is introduced that directly relates the performance of the fusion of individual sensor DCNN models to the fusion weight parameters. Using the introduced measure, we propose the minimum error score fusion (MESF) learning method that discriminatively optimizes the linear fusion weights of DCNN scores at the score level by minimizing the error rate on the training data in an iterative manner. The average gait identification accuracy of 97.06% is achieved by applying our proposed MESF method on the DCNN decision scores.

2. Method

We aim to automatically identify a target subject given their gait information. Assuming M target subjects, given an unknown gait segment, a gait identification system gives the corresponding subject identity, φ , where $\varphi \in \{1, \dots, M\}$. Figure 1 illustrates the structure of the gait identification task.

As shown in Figure 1, gait segments are processed and relevant features are extracted from them either manually or automatically. Then, a set of reference models, learned in the training phase using a set of training data, is employed to classify the input gait segments to one of the M subjects. Each model, φ , generates a likelihood score of an input gait segment belonging to the target subject φ . The aim is to identify each subject based on their individual gait characteristics given that all the subjects are performing the same activity i.e., gait. In order to achieved this goal in this paper, we capture and identify visual high level spectro-temporal features in an isolated gait cycle in a discriminative fashion using DCNN and multi-sensor fusion.

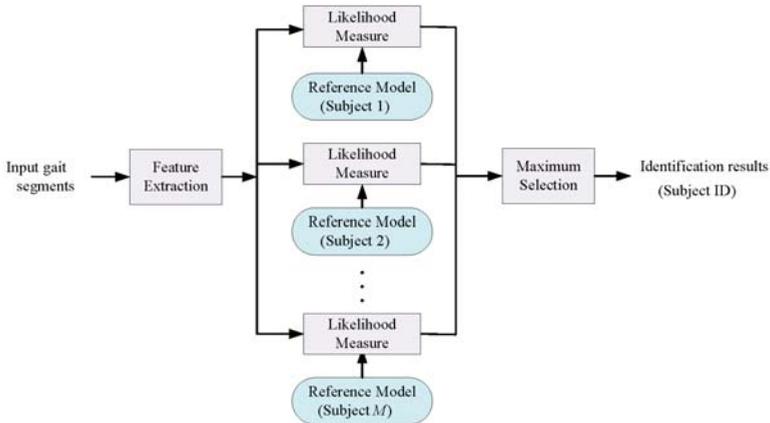


Figure 1. Human gait identification task.

The overview of the proposed gait identification system is depicted in Figure 2. Raw motion data is collected from five inertial sensors worn by a population of subjects. Then, gait cycles are extracted and transformed to 2D TF space. The high-level one-vs-rest subject discriminative features of the expanded gait cycles are captured through the 10-layer hierarchies of the DCNN and predictive model training is conducted jointly using the last layers of the DCNN network to perform the human identification task. We then combine the individual sensor systems via performing multi-sensor early and late fusion.

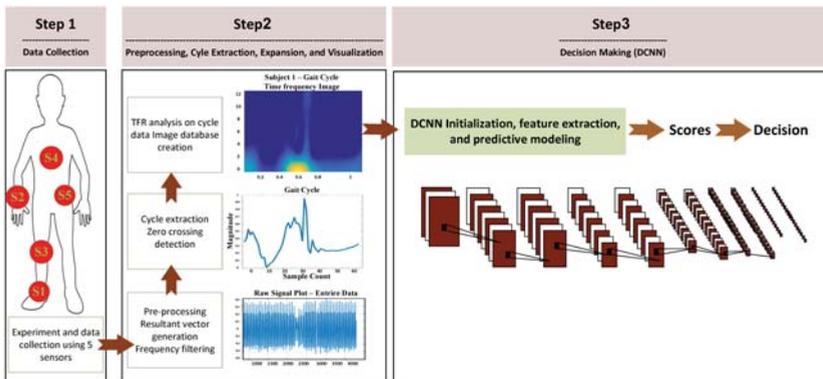


Figure 2. The overview of our proposed system for Human Gait Identification.

2.1. Experimental Setup and Data collection

A total of 10 subjects participated in the experimental procedure. In this way, we present a solution for a 10-class problem as a proof of concept of identifying a target person among overall 10 subject population (with 10% expected accuracy of random guess). A set of five inertial sensors were placed at various locations including chest, right wrist, right knee, right ankle and lower back in order to conduct a gait identification performance comparison between different sensor locations and improving the overall performance via multi-sensor fusion. Table 1 provides the detailed characteristics of the sensors [29,30]. The motion sensor system employed for this study was Shimmer sensor platform, which is a wearable sensor platform with wireless interface. It houses both the accelerometer and gyroscope inertial sensors. The data collection sessions were synchronized across all the sensors and labeled using our in-lab designed Android application developed at the Wearable Sensing and Signal Processing (WSSP) laboratory, University of Michigan, Dearborn. Each subject was asked to walk the same route from a specific point to another outside the building.

Table 1. The parameter values of the Gyroscope and Accelerometer sensors.

Gyroscope	Accelerometer
2 integrated dual-axis, InvenSense 500 series	3 Axis Accelerometer, Freescale MMA7260Q
Measures angular rate	Sensitivity: 800 mV/g @ 1.5 g
Full scale range: ± 5000 deg/s	12 bit analogue digital converter (integer number)
Sensitivity: 2 mV/deg/s	Resolution: $(1.5 \text{ g} + 1.5 \text{ g})/(2^{12}) \approx 7.10 - 4 \text{ g/unit}$
12 bit ADC (integer number)	-

2.2. Preprocessing

The raw accelerometer and gyroscope data (R_x , R_y and R_z vectors) collected during the experiment is contaminated with various noise factors such as motion artifacts, step impacts, sensor orientation and location related noises along with the necessary gait information. To alleviate the orientation related biases, resultant vectors of the triaxial sensor data (i.e., accelerometer and gyroscope recorded data) is computed using Equation (1). Figure 3 illustrates a visualization of resulting factor extraction, which is an orientation invariant measure of overall acceleration and angular velocity of each IMU.

$$Mag(t) = R = \sqrt{R_x^2(t) + R_y^2(t) + R_z^2(t)} \quad (1)$$

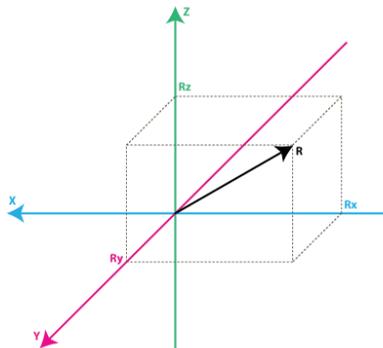


Figure 3. Resulting Factor Extraction. R indicates resulting factor.

Since typical gait data for normal walking has frequency components in the range of 0.5 to 3.5 Hz, a 10-th order *Butterworth* bandpass filter was used to extract the required frequency components from

the resultant vectors of the IMUs. For Gyroscope data, we assume the first estimated value of direction vectors to be the same as the direction vectors measured by the accelerometer:

$$\begin{aligned} R_x Est(0) &= R_x Acc(0) \\ R_y Est(0) &= R_y Acc(0) \\ R_z Est(0) &= R_z Acc(0) \end{aligned} \quad (2)$$

In our algorithm, we assume the value of the accelerometer, when the sensor device is at rest (a first few seconds recordings before subjects started the gait paradigm), to be zero.

2.3. Gait Cycle Extraction

Since the data collection through all the sensors is time synchronized, we extracted the gait cycles from the sensor#1 (i.e., the ankle sensor) and used the same markers for other sensors. The cycle extraction process implements amplitude check and zero crossing check to extract noise free gait cycle data.

To approximate the gait cycle frequency, the resultant vectors of accelerometer data from sensor#1 R_{Acc} are passed through a band pass filter of range 0.5–1.5 Hz. We conduct a more aggressive band pass filtering only for the purpose of cycle extraction. This eliminates the interference of any high frequency components while determining the gait cycle frequency. The frequency range of 0.5–1.5 Hz is chosen to include the average gait frequency, which is 1 Hz. Frequency Analysis consists of finding the energy distribution as a function of the frequency index ω . Therefore, it is necessary to transform the signal to the frequency domain by means of the Fourier transformation:

$$X(\omega) = \int x(t)e^{-jt\omega} dt \quad (3)$$

where $x(t)$ is the time domain signal. A Fourier transform is performed on the resultant data and the dominant frequency component within the range of 0.5 to 1.5 Hz is selected as the gait cycle frequency of that subject f_{cycle} .

In order to remove irregular gait cycles, the R_{Acc} is checked against the amplitude threshold of a signal 0.05. If there are samples that have values below the threshold, those samples are neglected and samples with an amplitude above the threshold are collected into small local windows. Gait cycles are extracted from these local windows (LW) following the process shown in the Algorithm 1.

Algorithm 1: Gait Cycle Extraction Process

```

 $S_{cycleStart} = 0$ 
while ( $LW_{end} < \text{length}(LW)$ )
     $S_{cycleEndTemp} = S_{cycleStart} + S_{cycleOffset}$ 
     $S_{cycleEnd} = S_{cycleEndTemp} \pm S_{zeroCrossing}$ 
     $S_{cycleStart} = S_{cycleEnd}$ 
     $LW_{end} = S_{cycleStart}$ 
end

```

To synchronize the cycles, zero crossing of the R_{Acc} is taken as a reference point and the samples between a given zero crossing and the subsequent zero crossing at $S_{cycleOffset}$ samples away from previous zero crossing is considered to be a gait cycle. $S_{cycleOffset}$ is calculated using Equation (4), where f_s is the data sampling frequency 50 Hz in our empirical investigations.

$$S_{cycleOffset} = \frac{f_s}{f_{cycle}} \quad (4)$$

The end sample of this cycle $S_{cycleEndTemp}$ is approximated initially based on the f_{cycle} but a more precise $S_{cycleEnd}$ is calculated based on finding a zero-crossing sample near the initial approximation. The local window indices of the start and end of cycle data $S_{cycleStart}$ and $S_{cycleEnd}$ are then mapped to the global window of R_{Acc} .

Cycle data collected across all subjects and sensors were observed for data consistency, i.e., the number of gait cycles extracted per sensor. Considering this number as a measure of data integrity, the ankle-based accelerometer (sensor#1) captured nearly all of the gait cycles performed by different subjects consistently. Hence, it was selected as the reference sensor with discriminative features to represent the gait cycle. Gait cycles are extracted from other sensors based on the indices generated from the sensor#1 R_{Acc} . Figure 4 shows the gait cycle extracted from Subject#1 for all the five sensors.

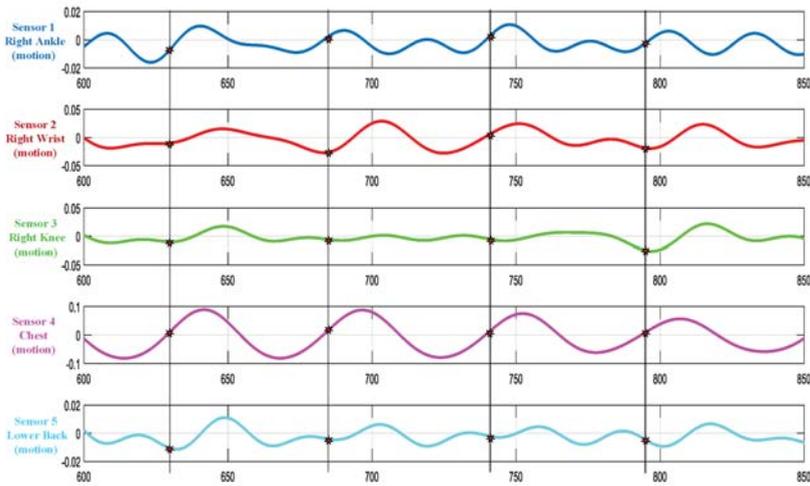


Figure 4. Extracted cycle data sample plot for subject 1 across all the sensors.

2.4. Time Frequency Representation

One way to represent and describe a multi-component and non-stationary signal simultaneously in frequency and time space is considering its instantaneous frequency (IF). To analyze a multi-component signal, an IF rule can be assigned to all components of the signal. Several IF estimation methods have been considered for multi-component signals in recent literatures [31]. These methods first characterize and extract components of the signal and then conduct an IF estimation procedure. Implementation of a multi-component IF estimation approach includes two major steps as follows:

1. Applying time frequency transformation.
First step is mapping the input signal to the time and frequency space by applying an appropriate time-frequency distribution (TFD). TFD method can be chosen by considering the characteristics of the input signal.
2. Choosing a proper method for estimating IF.
Methods for estimating IF consider the TFD space $(G(T, F))$ as a two dimensional representation, which its coordinates are row (time) and column (frequency) of the G space. Then, IF estimation method by applying first and second derivative tests identifies the local extremums (with respect to frequency). In this step, valid peak are the local extremums which have values higher or lower than a predefined threshold. Finally, for detecting the linked components an algorithm specifically designed for detection of linking component is applied by evaluating the connectivity

of the pixels and also the number of the connected pixels. The fact behind this is that IF of a component of a signal (where energy of the signal is concentrated) is observable in the TFD space as a ridge which describes the IF.

Selection of a proper TFD representation approach for representing gait cycle can be counted as the first step in designing any identification system using TF space. A proper TFD method is the one which is capable of emphasizing the non-stationarities of the given signal which, in turn, gives the system highest discriminative power to correctly discriminate between different cases in the population under consideration. In this study we use smoothed Wigner-Ville distribution (SWVD) as it is capable of reducing the cross-term affection while it provides good resolution [31,32]. SWVD is a variant method which incorporates smoothing by independent windows in time and frequency, namely (τ) and (t) :

$$SPWV(t, \omega) = \int_{-\infty}^{+\infty} W_{\omega}(\tau) \left[\int_{-\infty}^{+\infty} W_t(u-t) x(u + \frac{\tau}{2}) x^*(u - \frac{\tau}{2}) du \right] e^{j\omega\tau} d\tau \tag{5}$$

The feature extraction using SWVD is based on Energy, Frequency and Length of the principal track. Each segment gives the values E_k (energy), F_k (frequency), and L_k (length). The signal is firstly divided into segments; then, the construction of a three-dimensional feature vector for each segment will take place. Energy of each segment can be calculated as follows:

$$E_k = \int \int_{-\infty}^{+\infty} \vartheta_k(t, f) dt df \tag{6}$$

where $\vartheta_k(t, f)$ stands for the time-frequency representation of the segment. However, to calculate the frequency of each segment k , we made use of the marginal frequency as follows:

$$F_k = \int_{-\infty}^{+\infty} \vartheta_k(t, f) dt \tag{7}$$

For the purpose of SWVD representation we use a MATLAB toolbox designed by François Auger at CNRS (France) and Rice University (USA) [32]. Figure 5 illustrates the TF representation using 6 different TF approaches.

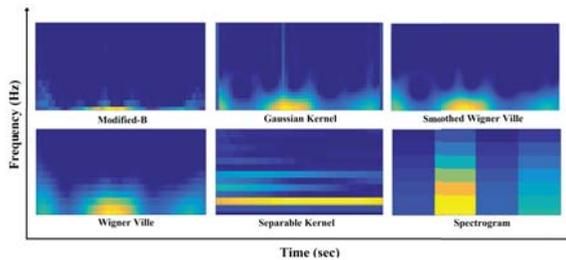


Figure 5. Time-frequency representation of one cycle data using different TFDs.

2.5. Deep Convolutional Neural Networks (DCNN)

Figure 6 demonstrates an example TF representation of gait cycle for 10 different subjects. Taking a close look at the gait cycle data, it is evident that the temporal orientation and order of the data stand out as significant discriminative features. We design a Convolutional Neural network (CNN) structure as a deep hierarchical structure for feature extraction and predictive modeling. We intend to verify if multiple feature maps generated by CNN process would preserve the temporal aspect of the gait cycle data and provide higher level of discriminative feature space.

CNNs are most commonly used in pattern recognition. A simple CNN is a sequence of steps where each step transforms a volume of activations through a set of functions that are differentiable. They are made up of neurons with learnable weights and biases [33]. Though the weight vector optimization is similar to the conventional neural networks. CNNs are designed to deal specifically with 2D or 3D image data. A variety of combinations of linear and non-linear differentiable steps could be used to build a deep CNN and that determines the complexity of the system [34].

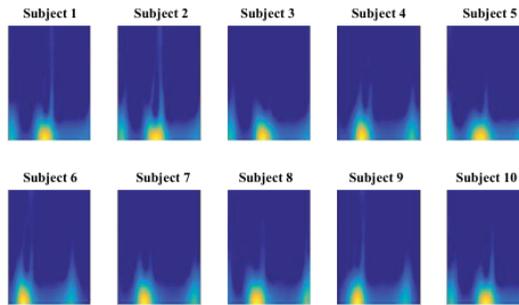


Figure 6. TF representation of gait samples as CNN inputs.

We have developed a non-parametric fully supervised DCNN model for motion-based gait authentication. The model takes a 3D input image x_i and transforms it into a prediction probability vector \bar{y}_i for ten different classes which correspond to the 10 participating subjects. We train the model using N labeled images $\{x, y\}$ where the label y_i is the class label of the input data. Training minimizes a *SoftMax* loss function with respect to network parameters such as weights and biases using a gradient descent method and network parameters are updated using back propagation.

We aim to conduct a comparison between the sensor locations in terms of how well each can describe subject dependent gait signatures towards the identification task. Furthermore, we aim to investigate multi-sensor fusion in order to enhance the gait identification performance using complementary information among various sensors.

DCNN Architecture

We have used the following four main building blocks in the DCNN model:

- Convolution.
- Pooling.
- Rectified linear unit (ReLU).
- Fully connected layer.

The convolution layer performs convolution of input with a set of predefined filters, which is a dot product between the filters and the region they are connected to in the input image. If we consider k kernels of spatial dimensions h_k and w_k , then the filter tensor would have the dimension of $h_k \times w_k \times k \times k \times k'$ which could be represented by tensor w . Considering a simple convolution layer with zero padding and a unit stride, the output y after performing convolution on an input layer of tensor x could be represented as

$$y_{i'j'k'} = \sum_{i=0}^{h_k} \sum_{j=0}^{w_k} \sum_{k'=0}^k w_{ijk'k} \times x_{i+i', j+j', k'} \quad (8)$$

In Equation (8), $x_{i+i', j+j', k'}$ represents the $(i+i', j+j', k')$ indexed element of the input tensor x .

The pooling layer applies a chosen operator and combines closely associated feature values. It is used to down sample the input image along the width and height. This layer does not require parameter learning. A simple implementation of max pooling can be represented as

$$y_{i'j'k} = \max_{0 \leq i \leq h_p, 0 \leq j \leq w_p} \{x_{i * h_p + i', j * w_p + j', k}\} \quad (9)$$

where x and y represent the $i'j'k$ indexed input and output layer and h_p, w_p are the pooling window dimensions.

The Rectified Linear Unit (ReLU) is a non-linear activation layer introduces the non-linearity when applied to the feature map. ReLU layer leaves the size of its input unchanged. A simple implementation of ReLU would be as below:

$$y_{i'j'k} = \max\{0, x_{i'j'k}\}. \quad (10)$$

where x and y are input and output of corresponding tensors. Like pooling layer, ReLU does not need any parameter learning and it does not alter the dimensions of the input layer.

3. Multi-Sensor Fusion

In this work, we investigate 5 different IMU sensor locations on the body and evaluate them for the gait identification task. Besides the comparison between the sensor locations in terms of how well each can distinguish subjects based on their gait, we also aim to investigate multi-sensor fusion in order to incorporate complementary information among various sensors to enhance the gait identification performance. The sensor fusion system is generally grouped into two types, namely early and late fusion. The basic idea of fusion is to combine multiple decisions generated by different experts in an attempt to enhance the performance of the overall system. A key issue to design a suitable and effective fusion scheme is to appropriately exploit all the available discriminative cues to generate an enhanced identification performance. Information fusion can be carried out at three levels of abstraction closely connected with the flow of the classification process: (1) data level fusion; (2) feature level fusion; and (3) model score fusion [35]. In this work, we present a simple data level fusion by concatenating the TFs from different sensor nodes and modalities into one collective input to the DCNN structure. We then propose and investigate an iterative minimum classification error multi-score fusion algorithm. Figure 7 demonstrates a block diagram of the proposed early and late fusion.

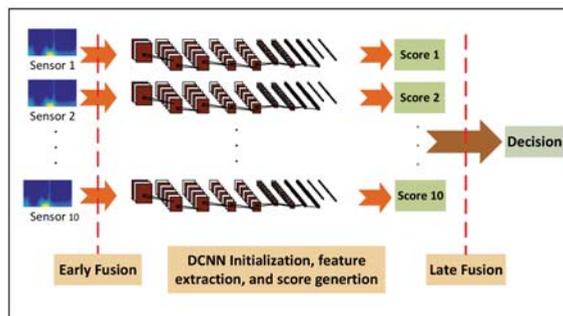


Figure 7. block diagram for early and late score fusion.

3.1. Early Fusion

Sensor fusion at the input level was first considered in this study. We aim to investigate multi-sensor fusion in order to enhance the gait identification performance using complementary information among various sensors. The key advantage of CNN is the feature learning ability,

which can automatically discover an intricate structure and learn useful features from raw data layer by layer, which enables CNN to fuse input raw data and extract basic information from it in its earlier layers, fuse the basic information into higher level representation of information, and further fuse those information in its later higher layers to form the final classification result. For early fusion, we aggregate the information in all the synchronous IMUs at the input level to feed the DCNN model. The input level fusion is achieved by combining the 3D gait cycle images from five different sensors to form a $120 \times 120 \times 30$ -dimensional image. The intuition behind early fusion is to incorporate all possible information that various sensors generate in order for the DCNN learning structure to learn the discriminative features in an iterative manner.

3.2. Late Fusion

The early-fusion method fuses two or more sensor readings by combining their transformed 2D gait cycle data. It provides a comprehensive input space to discover higher performance predictors. However, the input space grows in dimensions via early fusion and therefore, the search space for the best predictor increases exponentially. On the other hand, DCNN is a gradient descent learning method for the best predictor, which has serious limitations when the search space grows. Also, early fusion often cannot handle incomplete measurements. If one sensor modality becomes useless due to malfunctions, breakdown or etc, its measurements will be rendered ambiguous. In this section, we investigate late fusion of multiple sensor nodes (i.e., five inertial sensors) and modalities (i.e., Accelerometer and Gyroscope readings). In this way, a DCNN is independently trained to perform human gait identification based on each sensor location and modality. Subsequently, the one-vs.-rest discriminative output scores are fused at to generate the combined gait identification decision. We aim to design discriminative fusion of classification scores generated by individual DCNNs to reduce the classification error rate in the gait identification task. The fusion of multiple model scores operates as a mixture of experts that makes a collective decision by exploiting complementary information from each of the individual identification models. The input to each different classifier is generated by the TF spectro-temporal front-ends. If the model outputs associated with the different IMUs offer complementary information, multi-sensor fusion can improve the performance.

In this way, we generate the posterior probability scores from 10 DCNNs trained on the resulting factors of accelerometer and Gyroscope readings from five sensor nodes, individually. We introduce a discriminative performance measure to optimize the performance of the fusion of the 10 DCNNs. We propose the minimum error score fusion (MESF) method that iteratively estimates the parameters of a linear fusion with the aim to minimize the identification error rate on the training data. The proposed iterative algorithm learns the score weights one by one by finding the optimum decision threshold of each DCNN- i , $\{i | i = 1, \dots, N\}$ for each specific target subject.

3.3. The Minimum Error Score Fusion (MESF)

The human gait identification task is to recognize M target subjects using N individual gait classifiers (i.e., DCNNs). The i -th DCNN maps the gait segment \mathbf{x} to a score vector $S_i(\mathbf{x}) = \{S_{i,j}(\mathbf{x}) | j = 1, 2, \dots, M\}$, in which each element, $S_{i,j}(\mathbf{x})$, is the log-likelihood score of \mathbf{x} associated with the DCNN i and target subject j . The fusion output score for the target subject j , $Score_j(\mathbf{x})$, is calculated by combining the output scores of the DCNN sub-systems,

$$Score_j(\mathbf{x}) = \left\{ \sum_{i=1}^N w_{i,j} S_{i,j}(\mathbf{x}) \right\}, j = 1, 2, \dots, M \quad (11)$$

where $w_{i,j}$ is the weight corresponding to $S_{i,j}(\mathbf{x})$. However, unlike conventional linear score weighting techniques, the weighting coefficients are subject-dependent in which, the weighting coefficients vary for different subjects and individual DCNNs. By doing so, it is expected that the inter-subject discriminative information is taken into account. It also may reflect how one DCNN sub-system

contributes to identify each particular subject. There are a total of $N \times M$ weighting coefficients as fusion parameters to be estimated using the MESF algorithm. We then apply a normalization process on the scores of M target subjects.

3.3.1. The Process of Score Fusion

Figure 8 shows the process of learning the fusion weights. The MESF learning algorithm uses error feedback from the final fusion decision to fine-tune the fusion weights. We propose and employ a discriminative measure to find the optimum decision threshold in a one-vs-rest manner to learn the fusion weights of the general M -class identification problem. The gait recognition results are reported as the weighted average over multiple subject identifier DCNNs.

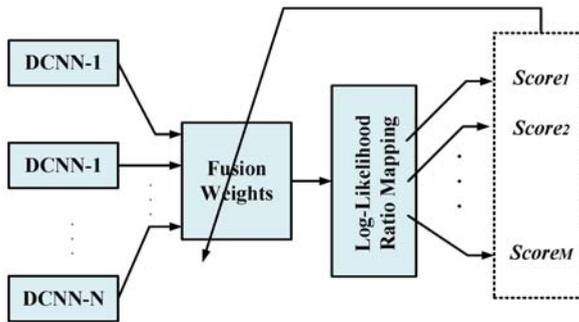


Figure 8. The training paradigm for learning the fusion weights.

As the distribution of the decision scores for each DCNN might be different due to different node locations and modalities, the scores are less compatible across different DCNN models. Hence, we conduct score normalization to provide consistency over the output scores of the DCNN models. The $Score_\varphi(\mathbf{x})$ from Equation (11) is converted to log-likelihood ratio (LLR) $\hat{Score}_\varphi(\mathbf{x})$ as presented in [36].

$$\hat{Score}_\varphi(\mathbf{x}) = Score_\varphi(\mathbf{x}) - \log \left\langle \frac{1}{M-1} \sum_{j=1, j \neq \varphi}^M \exp(Score_j(\mathbf{x})) \right\rangle \tag{12}$$

The input segment \mathbf{x} is classified as the target subject φ if,

$$\hat{Score}_\varphi(\mathbf{x}) > \theta_\varphi \tag{13}$$

where θ_φ is the decision threshold for the target subject φ , which is to be learned using cross validation on the training data. The weight $w_{i,\varphi}$, corresponding to the DCNNs output score $S_{i,\varphi}(\mathbf{x})$, can be interpreted as the degree to which $S_{i,\varphi}(\mathbf{x})$ contributes in the identification decision. From Equation (13), the fusion weight $w_{i,\varphi}$ can be directly related to the final decision as follows,

$$w_{i,\varphi} > \frac{\theta_\varphi + \log \left\langle \frac{1}{M-1} \sum_{j=1, j \neq \varphi}^M \exp(-Score_j(\mathbf{x})) \right\rangle - \sum_{k=1, k \neq i}^N w_{k,\varphi} S_{k,\varphi}}{S_{i,\varphi}(\mathbf{x})} \tag{14}$$

That is, $w_{i,\varphi}$ needs to satisfy Equation (14) in order for the input segment \mathbf{x} to be identified as the target subject φ . The MESF directly relates the performance of the system to the fusion parameters and determines the weights one at a time iteratively by solving a one-vs-rest subject identification problem to minimize the errors of the model score fusion.

3.3.2. The MESF Algorithm for Fusion Weight Learning

(1) Discriminative performance measure of the linear fusion:

DCNN models usually assign a score $S(x)$ to each unseen gait segment x , expressing the degree to which x would belong to each of the subject classes. Ideally, the score is an accurate estimate of posterior probability distributions over the target subject class (i.e., positive class), $p('pos'|x)$, and the rest of the subjects (i.e., negative class), $p('neg'|x)$, for an input x which denotes the estimated probabilities that x belongs to the positive and negative classes, respectively. We introduce a measure, $\Theta(x)$ to convert the probabilities into discriminative scores as below:

$$\Theta(x) = \frac{p('neg'|x)}{p('pos'|x)} \quad (15)$$

The measure $\Theta(x)$ demonstrates the the degree to which x is predicted to be of the negative class). By optimizing a threshold on the Θ measure, a statistical classifier can be converted to a discriminative classifier. In this way, the input x is classified as negative if the $\Theta(x)$ is greater than a specified threshold and positive otherwise. The desired threshold is learned by minimizing the classification error rate on the training data.

(2) Minimum error rate decision threshold in one-vs.-rest manner:

Assume that a training set $\{(x_t, y_t) | t = 1, 2, \dots, n\}$ consisting of n labeled data and their Θ measure $\Theta(x_t)$ is available. Using the training data, we aim to design an algorithm that finds the value of the linear fusion weight parameter that makes the optimum decision on a set of labeled training data samples. The proposed algorithm aims to return the decision threshold '*optimum_thresh*' that minimizes the error rate of the linear classifier fusion by varying the threshold value from 0 to $+\infty$ and find the optimum value by including false negatives (FN) in the positive class. Therefore, we rank the data samples in an ascending order of their $\Theta(\cdot)$ measure $\Theta(x_1), \dots, \Theta(x_{P+N})$. Considering any threshold, th , between $\Theta(x_t)$ and $\Theta(x_{t+1})$, the first K data samples will be classified as positive, where $\Theta(x_K) < th$, and the remaining $P + N - K$ data samples as negative. In this way, a maximum of $P + N + 1$ different thresholds has to be examined to find the optimum decision threshold. The first threshold classifies everything as negative and the rest of the thresholds are chosen in the middle of two successive Θ measures.

The threshold, *optimum_thresh*, on the $\Theta(\cdot)$ measure is found such that it minimizes the error rate of the DCNN model score fusion. The data sample x_t is classified as positive if $\Theta(x_t) \leq \textit{optimum_thresh}$. That is, x_t is classified as '*pos*' if,

$$\frac{p(x|neg')}{p(x|pos')} < \textit{optimum_thresh} \rightarrow p(x|neg') < \textit{optimum_thresh} \times p(x|pos') \quad (16)$$

The algorithm results in the highest accuracy achievable with the given $\Theta(\cdot)$ on the given set of data samples. In practice, the quality of the optimization depends on the quality of the positive and negative scores, e.g., the output posterior probability distributions of the DCNN models. In our fusion task, if the individual DCNN models generate good quality estimates, the fusion weight learning can considerably improve the performance. It can also be implemented in an efficient way by exploiting the monotonically ranked Θ measures.

(3) Discriminative learning of the fusion score weights

We propose an iterative learning procedure in order to minimize the error of the overall human identifier in one-vs-rest manner by re-estimating the weight parameters of the model score fusion. The proposed learning method adjusts the fusion weights in the interval $[0, \infty)$ using the training gait segments. The weights assigned to the output scores $S_{i,j}(\cdot)$ are set to one, for $w_{i,j} \leftarrow 1$ for $i = 1, \dots, N$, and $j = 1, \dots, M$, as an initial solution to the problem. Then, the following procedure is presented,

in which the error rate of the DCNN model fusion is successively reduced by finding a better solution than the current one by learning the locally optimum fusion weights, one at a time. To find the weight $w_{i,\varphi}$, corresponding to the output score of the DCNN i for target subject φ , the problem is considered to be a 2-class problem where the target subject φ is the positive class and $\bar{\varphi}$ comprising the rest of the subjects is the negative class. The steps are as follows:

1. The $w_{i,\varphi}$ is set to zero, i.e., $S_{i,\varphi}(\cdot)$ does not contribute in classification decision.
2. The gait segments in the training set belonging to the subject φ that are classified correctly with current values of the model score weights are marked. Adjusting the weight $w_{i,\varphi}$ does not have any impact on their classification.
3. The gait segments of $\bar{\varphi}$ in the training set that are misclassified are marked. Adjusting the weight $w_{i,\varphi}$ does not have any impact on their classification.
4. The remaining m gait segments in the training set $\{x_t | t = 1, \dots, m\}$ that are left unmarked are used to determine $w_{i,\varphi}$ so that the error of the model score fusion is minimized over $\{x_t | t = 1, \dots, m\}$. From Equation (14), the measure $\Theta_{i,\varphi}(\cdot)$ is calculated for every segment in $\{x_t | t = 1, \dots, m\}$ as follows:

$$\Theta_{i,\varphi}(x_t) = [\theta_\varphi + \log(\frac{1}{M-1} \sum_{j=1, j \neq \varphi}^M \exp(-Score_j(x_t))) - \sum_{k=1, k \neq i}^N w_{k,\varphi} S_{k,\varphi}(x_t)] / S_{i,\varphi}(x_t) \quad (17)$$

where, $\Theta_{i,\varphi}(x_t)$ is the amount of $w_{i,\varphi}$ necessary for x_t to be classified as the target subject φ .

5. The training segments are ranked in an ascending order by their $\Theta_{i,\varphi}(\cdot)$ measure. A threshold is defined and initialized to zero. Then, assuming that x_t and x_{t+1} are two successive segments in the list, the threshold is computed as,

$$thresh = [\Theta_{i,\varphi}(x_t) + \Theta_{i,\varphi}(x_{t+1})] / 2 \quad (18)$$

Then, the threshold is adjusted from the lowest score to the highest. For each threshold, the associated accuracy of the score fusion is measured. The value of the *optimum_thresh* leading to the maximum accuracy is used as the estimated score weight $w_{i,\varphi}$, assuming that all other score weights are fixed.

Figure 9 shows a data sample of a resulting threshold on the $\Theta_{q,w}(\cdot)$ measure leading to the highest accuracy. In this way, the input x_t is classified as w if its $\Theta(x_t)$ is lower than the threshold. The algorithm determines the optimal value of the parameter $\phi_{q,w}$ assuming that all other parameters are given and fixed.

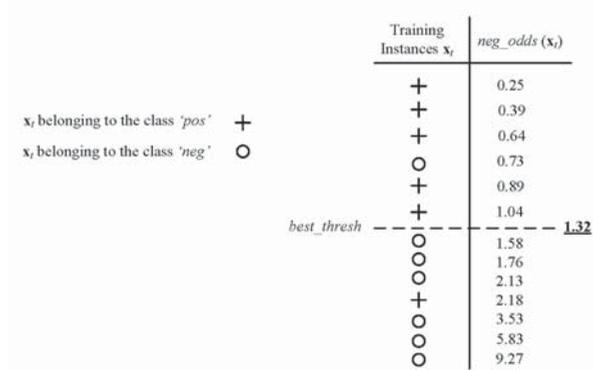


Figure 9. A data sample of the resulting thresholds on the $\Theta_{q,w}(\cdot)$ measure.

The search for the locally optimum combination of weights is conducted by optimizing the score weights one at a time and learning stops if no improvement to the current performance can be made. Based on our observations, the above method consistently generated superior results. The MESF method determines the decision score weights of the sub-systems in order to better discriminate between the gait segments from subjects φ and those of the rest by finding the optimum decision threshold of the score fusion in the one-vs-rest manner. By doing so, we can interpret that the ROC curve of the score fusion is locally optimized. As shown below, it is illustrated that the error rate on the training data always reduces during the optimization and will converge to a minimum point.

$$Error_{Rate}(\Lambda \cup \Phi_{q,w}^{new}, D) \leq Error_{Rate}(\Lambda \cup \Phi_{q,w}^{old}, D) \tag{19}$$

where Λ is the set of all other parameters (Given & Fixed)

$$\Rightarrow Error_{Rate}(\Lambda \cup \Phi_{q,w}^t, D) \leq Error_{Rate}(\Lambda \cup \Phi_{q-1,w}^{t-1}, D) \geq Error_{Rate}(\Lambda \cup \Phi_{q-2,w}^{t-2}, D) \leq \dots \tag{20}$$

where t represent the iteration number starting from 1 and increment after each learning iteration. After each iteration, the accuracy of the learning algorithm decreases.

3.4. Comparable Score Fusion Methods

In this section, some other linear score fusion methods are described for comparison. Assume an input gait segment \mathbf{x} , and the output decision scores of the sub-systems, $\{S_{i,j}(\mathbf{x}) | i = 1, \dots, N \ \& \ j = 1, \dots, M\}$ are available. Comparative methods include a simple non-trainable combiner and the large margin method that require a more sophisticated training procedures.

- Summation: The output scores of the individual DCNNs are summed up and the subject that receives the highest score is the output decision of the fusion system.

$$Score_j^{sum}(\mathbf{x}) = \sum_{i=1}^N S_{i,j}(\mathbf{x}) \tag{21}$$

- Support vector machine (SVM): SVM has shown to be effective in separating input vectors in 2-class problems [37], in which SVM effectively projects the vector \mathbf{x} into a scalar value $f(\mathbf{x})$,

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + d \tag{22}$$

where the vectors \mathbf{x}_i are support vectors, $y_i = \{-1, 1\}$ are the correct outputs, N is the number of support vectors, (\cdot) is the dot product function, α_i are adjustable weights, and d is a bias. Learning is posed as an optimization problem with the goal of maximizing the margin, which is the distance between the separating hyperplane and the nearest training vectors. As a combiner method, the output scores of all the DCNNs are concatenated to form a vector as the input to SVMs. Then, one SVM is assigned to each of the M target subjects, the one-vs-rest scheme, and train accordingly to get the final output scores,

$$Score_j^{SVM}(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i (\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x})) + d \tag{23}$$

where $\phi(\mathbf{x})$ is the vector resulted by concatenating the output scores of all the DCNNs for the input segment \mathbf{x} .

4. Results

We implemented a Matlab TF representation toolbox [38] to generate the SWVD TF representation of the data and use Matlab print function to output a 2D image with specific resolution and size. After experimenting with multiple image sizes, it was observed that higher image resolution did not necessarily mean better model performance. DCNN model was tested with different input image sizes. The image size was modified by changing the resolution of the figure generated by Matlab print function. It was observed that the model prediction accuracies are highest with the image size of $120 \times 120 \times 3$ for our current analysis.

For DCNN model designed in this study, the architecture guidelines as mentioned in [39,40] were followed. Small filters were used and the image size was reduced using higher stride lengths where necessary. Padding was introduced in the convolution layers to prevent convolution layer from altering the spatial dimensions of the input. The spatial dimensions were altered in pooling layer by down sampling. Also, more ReLU activation layers were used across the DCNN after each convolution and pooling pair to bring in element wise non-linearity. The current model consists of 10 (CONV, RELU, POOL) layers and one fully connected layer as shown in Figure 10.

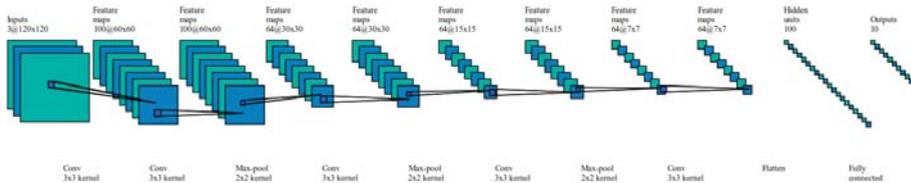


Figure 10. CNN architecture.

4.1. Model Selection and Evaluation Procedure

We consider the human gait identification task as a pattern recognition problem, in which a set of training samples was employed for gait signature pattern extraction and predictive model training, a separate set of validation samples was used for hyperparameter tuning, and an exclusive set of test samples was used for predictive model evaluation. In order to estimate the generalization accuracy of the predictive models on the unseen data, 10 fold cross validation (10-CV) was used. 10-CV divides the total input data of n samples into ten equal parts. In every iteration one part is considered to be a test sample set and the remaining nine parts are considered to be a validation and training sample set. There is no overlap between the test sample set (10% of data) with the validation and training sample set (90% of data). The latter set is further divided into 4:1 ratio of training and validation data samples. The sets were permuted over 10 iterations to generate an overall estimate of the generalization accuracy. For an input data set of n samples:

- Number of Test Samples: $n/10$
- Number of Validation samples: $(n - \frac{n}{10}) \times 1/5$
- Number of Training samples: $(n - \frac{n}{10}) \times 4/5$.

Data set has $n = 4178$ samples, which includes images from ten different subjects. Each iteration of the 10—fold cross validation would have the following division of data (Note that there was no overlap between the training set, the validation set, and the test set in all the iterations of 10-CV).

- Number of Test Samples: 417
- Number of Validation samples: 752
- Number of Training samples: 3008

Figure 6, shows a sample image set of all subjects for sensor#1. The DCNN model was trained using the training and validation set and tested independently with the testing set. In order to fine-tune the parameters of the DCNN model, we conducted random restart hill climbing search method on a low resolution quantized parameter space to achieve a reliable local minimum in the error function (i.e., the average error rate of the model on the 5 individual sensor locations). Table 2 reports the selected parameters to train the gait identification DCNN models. It must be noted that no of epochs was set to 19 since, we terminate training at 19 epochs where we get maximum accuracy and the model is able to generalize enough and avoids overfitting.

Table 2. CNN predefined parameters.

Parameter	Values
Learning Rate	0.001
Momentum Coefficient	0.9
No. of Feature Maps	32, 64
No. of Neurons in Fully Connected Layer	64
Batch Size	40
Epoch Number	19
Epoch Number	19

4.2. Individual Sensor Performance

The prediction accuracies of the DCNN models for each iteration of the 10-CV and for different sensors are shown in Figure 11. It demonstrates the average accuracies achieved using each of the five individual sensors over the 10 iterations of 10-CV. The blue and red markers represent the accuracies corresponding to the accelerometer and gyroscope data recorded from each sensors, respectively. It can be observed that the gyroscope generated data (i.e., angular velocity) of the sensors are better predictors than the acceleration data in 3 out of 5 cases. Only the acceleration data from sensor#1 (on the right ankle) and sensor#2 (on the right wrist) exhibited higher gait identification accuracy compared to angular velocities from the same sensors. It might be due to wider range movements of the ankles and wrists generating more individually descriptive accelerations. The acceleration data from sensor#1 on the ankle generates the best average 10-CV accuracies among all 10 different sources of data (Acc No.). This observation was expected since dynamics of gait is mostly captured by the lower limb movement [41]. For the gyroscope data, the average 10-CV prediction accuracy for different sensors in decreasing order is as follows: sensor#5 (lower back), sensor#3 (right knee), sensor#4 (chest), sensor#1 (right ankle), sensor#2 (right wrist). For the accelerometer data, prediction accuracies of the sensors in descending order is: sensor#1, sensor#3, sensor#5, sensor#2, sensor#4. The ranking of the sensors in terms of identification accuracies are not consistent for gyroscope and accelerometer sensors. It loosely suggests that upper trunk locations is more suitable for Gyroscopes and lower limb locations are more suitable for accelerometers. The accelerometer at the at the right ankle sensor (capturing high range lower limb acceleration) and the gyroscope sensor of of the lower back sensor (capturing the angular velocity of trunk movements) generated the highest accuracies. This observation suggests that incorporating the complementary discriminative information generated from different sensors and modalities can be fused and improved. Therefore, multi-sensor fusion is investigated in early and late fashions in order to improve the the recognition performance.

4.3. Early Fusion

In early fusion, we investigate aggregating the complementary discriminative motion data recorded synchronously from the five IMUs. In this way, we conduct a sensor fusion at the input level to the DCNN model as discussed in Section 3.1. The input level fusion is achieved by combining the 3D gait cycle images from five different sensors to form a $120 \times 120 \times 30$ -dimensional image. The average subject identification accuracy percentages of the DCNN model, whose parameters are optimized

using 10-CV cross validation method, is shown in the Figure 12. The early fusion via input image aggregation at the input to one overall DCNN demonstrated improved prediction accuracies as shown in Figure 12. Results report that using Max pooling is preferable over Min pooling leading to higher identification accuracy in 8 out of 10 subject cases. The average 10-CV gait identification performance achieved, using Max pooling, was 93.36%. The results suggests that using a simple input aggregation fusion of sensors by aggregation can enhance the performance of the best DCNN model (i.e., Sensor#1 Acc with 91.01% accuracy) with 26.2% relative improvement. In order to compare the results of the early fusion model performance and Sensor#1 Gyro DCNN model, we conducted 10-CV test 10 times and run a statistical test with the null hypothesis that the performance accuracies of the fusion and 'S1 Acc' on different subjects comes from independent random samples from normal distributions with equal means, using the two-sample t-test. The test rejected the null hypothesis with the p -value = 0.028 which demonstrate significant improvement over the best individual sensor-based DCNN model under the 95% confidence interval.

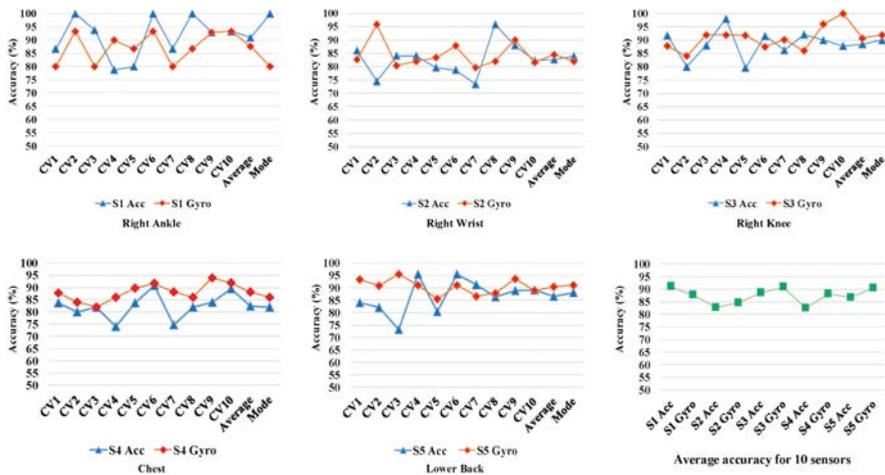


Figure 11. 10 Cross Validation Performance of the Individual Sensors Including Both Gyroscope (Gyro) and Accelerometer (Acc) sensors. CV_j indicates Cross-Validation j , and S_i indicates sensor i .

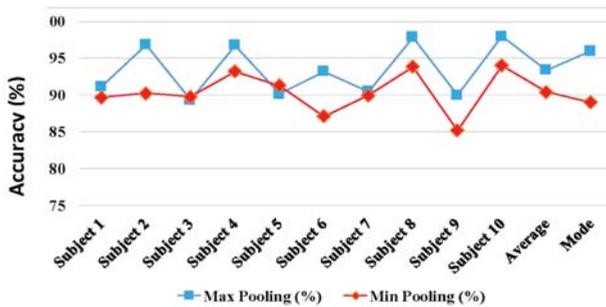


Figure 12. Multi-sensor early fusion performance using min and max pooling.

4.4. Late Fusion

For late fusion, we employ the MESF algorithm described in Section 3.3. The discriminative measure to find the optimum decision threshold in one-vs-rest manner introduced in Section 3.2 is

utilized to learn the fusion weights of the general M -class identification problem ($M = N = 10$). Therefore, the total number of weight coefficients as fusion parameters to be estimated are $N \times M = 50$. We applied log likelihood score normalization in order to provide consistency over the output scores of the individual DCNNs. Table 3 presents the results for score level fusion using the MESF algorithm. Table 3 reports that MESF using Max pooling lead to higher identification accuracy compared to Min pooling in 9 out of 10 subject cases. As Table 3 reports, the proposed MESF method generates the average 10-CV accuracies of 97.06% and 95.24% using Max pooling and Min pooling, which lead to 67.3% and 47.1% relative improvement compared to the best individual DCNN results, respectively.

Table 3. MESF, SUM, and SVM performance using min and max pooling. Bold numbers represent the highest accuracy achieved for each subject using different methods.

	MESF		SUM		SVM	
	Max (%)	Min (%)	Max (%)	Min (%)	Max (%)	Min (%)
Subject 1	95.56	94.61	89.93	89.45	92.31	90.15
Subject 2	100	97.44	92.41	89.26	96.87	91.3
Subject 3	92.91	94.26	88.97	88.35	89.12	91.65
Subject 4	99.35	97.38	93.82	90.7	94.71	90.26
Subject 5	95.76	95.56	89.13	89.29	92.34	90.89
Subject 6	97.27	93.6	93.62	89.44	92.49	89.41
Subject 7	96.15	93.12	87.28	86.31	91.2	88.54
Subject 8	98.93	97.63	92.36	89.04	95.34	92.81
Subject 9	94.65	90.05	87.94	85.75	90.57	87.31
Subject 10	100	98.78	94.2	91.37	97.69	95.95
Average	97.06	95.24	90.96	88.89	93.27	90.83

Using the MESF fusion algorithm, we achieved the highest 10-CV accuracy of 97.06% on the 10-class identification problem with 10% expected accuracy of random guess. In the case of having a larger subject population, its impact will be solely on the number of impostors per each one-vs-rest subject identifier (the number of targets will remain the same). That will limit the potentially introduced errors to false alarms. On the other hand, our proposed method in this paper is designed to minimize the one-vs-rest identification errors. By decoding the gait cycles via spectro-temporal expansion, we discover and capture discriminative individual low and high level signatures and predictive model jointly via the trained DCNN structure and then discriminatively multi-sensor fusion model. Our aim with the error minimizing design of joint our proposed models was to equip them to cope with both false alarms and miss detections.

4.5. Comparative Study

We also compare the proposed MESF method with two other linear score fusion methods, namely SUM and SVM methods introduced in Section 3.4. Table 3 presents the results for our MESF method, SVM, and SUM using Max and Min pooling. According to Table 3, Max pooling performs better in majority of cases for various subjects and fusion methods. SUM method did not demonstrate improvements in either cases of Max and Min pooling compared to the best individual DCNN. It might be due to the fact that the lower accuracy DCNN model scores adversely affect the fusion identification decisions. On the other hand, SVM fusion method generates improved results in case of Max pooling via solving an optimization problem with the goal of maximizing the margin between the target subjects and the others. In this way, one SVM is assigned to each of the 10 target subjects (i.e., one-vs.-rest scheme). We applied the two-sample t-test to evaluate the significance of improvement of SVM fusion compared to the best individual DCNN. The test rejected the null hypothesis with the p -value = 0.034, which is significant under 95% confidence level. Although the performances of the SVM fusion relatively improves the best individual performance by 25.2%, they are inferior to the MESF performance.

5. Conclusions

In the human gait recognition task, the aim is to extract discriminative features and descriptors from gait motion signals to identify our target subject from others. The manual feature extraction is prone to error due to the complexity of data collected from inertial sensors and the disconnection between feature extraction and the discriminative learning models. To overcome this shortcoming, we proposed a novel methodology for processing non-stationary signals for the purpose of human gait identification. The proposed methodology comprises four main components: (1) cycle extraction; (2) spectro-temporal 2D expansion and representation; (3) deep convolutional learning; and (4) discriminative multi-sensor model score fusion. We first isolated the gait cycles using a simple and effective heuristic. Then, we conducted spectro-temporal transformation of isolated gait cycles. The 2-D expansion of the gait cycle increased the resolution of the desired discriminative trends in the joint time-frequency domain. In order to avoid manual feature extraction and incorporate joint feature and model learning from the generated high resolution 2D data, we designed a deep convolutional neural network structure in order to process the signal layer by layer, extract discriminative features, and jointly optimize the features and the predictive model via error back propagation model training. We investigated 5 IMU sensor placement on the body and conducted a comparative investigation between them in terms of gait identification performance using synchronized gait data recordings from 10 subjects. Due to complementary discriminative signature patterns captured from the recorded signals collected via different sensors, we then investigated gait identification fusion modeling from multi-node inertial sensor data and effectively expand it via 2-D time-frequency transformation. We perform early (input level) and late (model score level) multi-sensor fusion to improve the cross validation accuracy of the gait identification task. We particularly proposed the minimum error DCNN model score fusion algorithm. Based on our experimental results, 93.88% and 97.06% subject identification accuracy was achieved via early and late fusion of the multi-node sensor readings, respectively.

6. Future Research Direction

In this paper, the problem of human gait identification was investigated under the bigger umbrella of ubiquitous and continuous IMU-based gait analysis work group at the Wearable Sensing and Signal Processing (WSSP) lab. Our main contribution and focus in this paper was our proposed model score fusion algorithm to incorporate the complementary discriminative scores generated by each individual DCNN model given a input human gait cycles. We used a local search and an overall accuracy to generate a reliable baseline DCNN model for all the IMU recordings. Due to the fact that data from different sensors (i.e., sources) may have different characteristics, the DCNN training paradigm can be improved by sensor-dependent tuning for different sensor locations and modalities. Therefore, our team is currently investigating sensor location and modality specific DCNN model optimization and subsequently, designing multi-sensor and multi-model fusion algorithms. We are also increasing the number of subjects and the walking conditions in our data set. When completed, our plan is to create a link on our website to share our data set and our baseline model implementations with the research community.

Acknowledgments: The authors gratefully acknowledge financial support from Ford smart mobility center.

Author Contributions: Omid contributed in concept of the study, CNN architecture design and discriminative fusion algorithm, writing and preparing the manuscript. Mojtaba contributed in data analytics and time—frequency representation, CNN design and result generation, and writing and preparing the manuscript. Raghavendar contributed in gait cycle extraction, conducting CNN and fusion result generation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Murray, M.; Pat, A.; Bernard, D.; Ross, C.K. Walking patterns of normal men. *JBS* **1964**, *46*, 335–360.

2. Jellinger, K.; Armstrong, D.; Zoghbi, H.Y.; Percy, A.K. Neuropathology of Rett syndrome. *Acta Neuropathol.* **1988**, *76*, 142–158.
3. Katz, J.N.; Dalgas, M.; Stucki, G. Degenerative lumbar spinal stenosis Diagnostic value of the history and physical examination. *Arthritis Rheumatol.* **1995**, *38*, 1236–1241.
4. Nutt, J.G.; Marsden, C.D.; Thompson, P.D. Human walking and higher-level gait disorders, particularly in the elderly. *Neurology* **1993**, *43*, 268.
5. El-Sheimy, N.; Haiying, H.; Xiaoji, N. Analysis and modeling of inertial sensors using Allan variance. *IEEE Trans. Instrum. Meas.* **2008**, *57*, 140–149.
6. Kale, A.; Sundaresan, A.; Rajagopalan, A.N.; Cuntoor, N.P.; Roy-Chowdhury, A.K.; Krüger, V.; Chellappa, R. Identification of humans using gait. *IEEE Trans. Image Process.* **2004**, *13*, 1163–1173.
7. Sprager, S.; Matjaz, B.J. Inertial sensor-based gait recognition: A review. *Sensors* **2015**, *15*, 22089–22127.
8. Gafurov, D.; Einar, S.; Patrick, B. Gait authentication and identification using wearable accelerometer sensor. In Proceedings of the IEEE Workshop on Automatic Identification Advanced Technologies, Alghero, Italy, 7–8 June 2007.
9. Kim, E.; Sumi, H.; Diane, C. Human activity recognition and pattern discovery. *IEEE Pervasive Comput.* **2010**, *9*, 48.
10. Mortazavi, B.; Alsharufa, N.; Lee, S.I.; Lan, M.; Sarrafzadeh, M.; Chronley, M.; Roberts, C.K. MET calculations from on-body accelerometers for exergaming movements. In Proceedings of the IEEE International Conference on Body Sensor Networks (BSN), Cambridge, MA, USA, 6–9 May 2013.
11. Vikas, V.; Carl, D.C. Measurement of robot link joint parameters using multiple accelerometers and gyroscopes. In Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, OR, USA, 4–7 August 2013.
12. Robertson, K.; Rosasco, C.; Feuz, K.; Cook, D.; Schmitter-Edgecombe, M. C-66 Prompting Technologies: Is Prompting during Activity Transition More Effective than Time-Based Prompting? *Arch. Clin. Neuropsychol.* **2014**, *29*, 598.
13. Ahmadi, A.; Edmond, M.; Francois, D.; Marc, G.; Noel, E.O.; Chris, R.; Kieran, M. Automatic activity classification and movement assessment during a sports training session using wearable inertial sensors. In Proceedings of the International Conference on IEEE Wearable and Implantable Body Sensor Networks (BSN), Zurich, Switzerland, 16–19 June 2014.
14. Le, M.J.; Ingo, S. The smartphone as a gait recognition device impact of selected parameters on gait recognition. In Proceedings of the 2015 International Conference on Information Systems Security and Privacy (ICISSP), Angers, France, 9–11 February 2015.
15. Wang, Y.; Kobsa, A. *Handbook of Research on Social and Organizational Liabilities in Information Security, Chapter Privacy Enhancing Technology*; Idea Group Inc. Global: Miami, FL, USA, 2008.
16. Chen, C.; Roozbeh, J.; Nasser, K. A survey of depth and inertial sensor fusion for human action recognition. *Multimed. Tools Appl.* **2017**, *76*, 4405–4425.
17. Roberts, M.L.; Debra, Z. *Internet Marketing: Integrating Online and Offline Strategies*; Cengage Learning: Boston, MA, USA, 2012.
18. Yamada, M.; Aoyama, T.; Mori, S.; Nishiguchi, S.; Okamoto, K.; Ito, T.; Muto, S.; Ishihara, T.; Yoshitomi, H.; Ito, H. Objective assessment of abnormal gait in patients with rheumatoid arthritis using a smartphone. *Rheumatol. Int.* **2012**, *32*, 3869–3874.
19. Sposaro, F.; Gary, T. iFall: An Android application for fall monitoring and response. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, USA, 3–6 September 2009.
20. Yurtman, A.; Billur, B. Activity Recognition Invariant to Sensor Orientation with Wearable Motion Sensors. *Sensors* **2017**, *17*, 1838.
21. Mantyjärvi, J.; Lindholm, M.; Vildjounaite, E.; Makela, S.-M.; Ailisto, H.A. Identifying users of portable devices from gait pattern with accelerometers. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), Philadelphia, PA, USA, 23–23 March 2005.
22. Subramanian, R.; Sudeep, S.; Miguel, L.; Kristina, C.; Christopher, E.; Omar, J.; Jiejie, Z.; Hui, C. Orientation invariant gait matching algorithm based on the Kabsch alignment. In Proceedings of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2015), Hong Kong, China, 23–25 March 2015.

23. Gafurov, D.; Einar, S.; Patrick, B. Improved gait recognition performance using cycle matching. In Proceedings of the 24th International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia, 20–23 April 2010.
24. Rong, L.; Duan, Z.; Zhou, J.; Liu, M. Identification of individual walking patterns using gait acceleration. In Proceedings of the 2007 1st International Conference on Bioinformatics and Biomedical Engineering, Wuhan, China, 6–8 July 2007.
25. Kwapisz, J.R.; Gary, M.W.; Samuel, A.M. Cell phone-based biometric identification. In Proceedings of the 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), Washington, DC, USA, 27–29 September 2010.
26. Thang, H.M.; Vo Quang, V.; Nguyen, D.T.; Deokjai, C. Gait identification using accelerometer on mobile phone. In Proceedings of the 2012 International Conference on Control, Automation and Information Sciences (ICCAIS), Ho Chi Minh City, Vietnam, 26–29 November 2012.
27. Sprager, S.; Matjaz, B.J. An efficient HOS-based gait authentication of accelerometer data. *IEEE Trans. Inform. Forensics Secur.* **2015**, *10*, 1486–1498.
28. Hrushikesh, M.; Qianli, L.; Tomaso, P. When and Why Are Deep Networks Better Than Shallow Ones? In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
29. SHIMMER Getting Started Guide. Available online: <http://www.eecs.harvard.edu/~konrad/projects/shimmer/SHIMMER-GettingStartedGuide.html> (accessed on 27 November 2017).
30. Baharev, A. Gyroscope Calibration. Available online: <http://reliablecomputing.eu/baharev-gyro.png> (accessed on 27 November 2017).
31. Boashash, B. Estimating and interpreting the instantaneous frequency of a signal. II. Algorithms and applications. *Proc. IEEE* **1992**, *80*, 540–568.
32. Pedersen, F. *Joint Time Frequency Analysis in Digital Signal Processing*; Aalborg Universitetsforlag: Aalborg, Denmark, 1997.
33. Karpathy, A. Cs231n: Convolutional Neural Networks for Visual Recognition. *Neural Netw.* **2016**. Available online: <http://cs231n.github.io/> (accessed on 27 November 2017).
34. Zhong, Y.; Yunbin, D. Sensor orientation invariant mobile gait biometrics. In Proceedings of the IEEE International Joint Conference on Biometrics, Clearwater, FL, USA, 29 September–2 October 2014.
35. Bezdek, J.C. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
36. Reynolds, D.A.; Thomas, F.Q.; Robert, B.D. Speaker verification using adapted Gaussian mixture models. *Digit. Signal Process.* **2000**, *10*, 19–41.
37. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
38. Auger, F.; Patrick, F.; Paulo, G.; Olivier, L. *Time-Frequency Toolbox*; CNRS France-Rice University: Paris, France, 1996.
39. Tong, M.; Minghao, T. LEACH-B: An improved LEACH protocol for wireless sensor network. In Proceedings of the 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, China, 23–25 September 2010.
40. Rigamonti, R.; Matthew, A.B.; Vincent, L. Are sparse representations really relevant for image classification? In Proceedings of the 2011 IEEE Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011.
41. Dehzangi, O.; Zheng, Z.; Mohammad-Mahdi, B.; John, B.; Christopher, R.; Roozbeh, J. The impact of vibrotactile biofeedback on the excessive walking sway and the postural control in elderly. In Proceedings of the 4th Conference on Wireless Health, Baltimore, MD, USA, 1–3 November 2013.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Heterogeneous Data Fusion Method to Estimate Travel Time Distributions in Congested Road Networks

Chaoyang Shi ^{1,2,3}, Bi Yu Chen ^{1,2,3,*}, William H. K. Lam ³ and Qingquan Li ^{1,2,4,*}

¹ State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China; chaoyangshi@whu.edu.cn

² Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China

³ Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China; william.lam@polyu.edu.hk

⁴ Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen 518060, China

* Correspondence: chen.biyu@whu.edu.cn (B.Y.C.); qqli@whu.edu.cn (Q.L.); Tel.: +86-27-6877-9771 (B.Y.C.); +86-755-2697-9741 (Q.L.)

Received: 2 November 2017; Accepted: 5 December 2017; Published: 6 December 2017

Abstract: Travel times in congested urban road networks are highly stochastic. Provision of travel time distribution information, including both mean and variance, can be very useful for travelers to make reliable path choice decisions to ensure higher probability of on-time arrival. To this end, a heterogeneous data fusion method is proposed to estimate travel time distributions by fusing heterogeneous data from point and interval detectors. In the proposed method, link travel time distributions are first estimated from point detector observations. The travel time distributions of links without point detectors are imputed based on their spatial correlations with links that have point detectors. The estimated link travel time distributions are then fused with path travel time distributions obtained from the interval detectors using Dempster-Shafer evidence theory. Based on fused path travel time distribution, an optimization technique is further introduced to update link travel time distributions and their spatial correlations. A case study was performed using real-world data from Hong Kong and showed that the proposed method obtained accurate and robust estimations of link and path travel time distributions in congested road networks.

Keywords: travel time distribution; data fusion; evidence theory; spatial correlation; uncertainty

1. Introduction

Accurate and robust estimation of travel time distribution, including mean and variance, is a crucial requirement for advanced traveler information systems (ATIS). Provision of travel time distribution information through ATIS enables travelers to make reliable path choice decisions, ensuring a higher chance of on-time arrival [1–3]. The provided distribution information also allows operators to evaluate network performance and reliability, and identify bottlenecks for proactively deploying effective controls to improve overall traffic conditions [4,5].

Recent advances in information and communication technologies (ICTs) have produced a variety of spatiotemporal big data for travel time estimation [6]. Existing data collection techniques could be classified into point detection, interval detection, and floating car systems [7,8]. Point detectors (such as loop detectors and video image detectors) are generally deployed at specific road segment locations, to collect vehicle point speeds. Interval detectors consist of a pair of devices deployed in road networks to directly calculate travel times between the device pair. Typical interval detectors include automatic vehicle identification (AVI), Bluetooth, and license plate recognition devices. In contrast

to the above fixed detectors, floating car systems use a fleet of probe vehicles, typically taxi cabs, equipped with global positioning system (GPS) devices. The probe vehicle locations and speeds are collected at given time intervals to estimate network traffic conditions [9]. These data collection techniques have generated complementary heterogeneous data sources with distinct data quality and network coverage.

Accurate and robust estimation of travel time distributions from heterogeneous data sources is somewhat challenging in congested road networks. Firstly, although rich traffic observations from multiple data sources are beneficial, data quality variability from different data sources presents a serious impediment to robust estimation of travel time distributions. Data quality variability may raise from a variety of reasons, such as detector failures, measurement errors, sample size variations, etc. [10–13]. Therefore, traffic observations from different sources can be inconsistent and even conflicting. Thus, robust data fusion techniques are urgently required, relatively insensitive to data quality of heterogeneous data sources.

Secondly, traffic data from multiple data sources has enhanced data availability for major roads in a network, but the limited coverage across the whole network poses a significant challenge to accurate travel time distribution estimates. Traffic data from point detectors cover all vehicles at deployed locations and have a very good temporal sampling, but their spatial coverage is restricted to the (relatively few) deployed locations. Floating car and interval detector data have relatively better spatial coverage on major roads, but sparse data issues remain for many arterial roads [14,15]. Therefore, effective techniques to impute spatially missing data are also required.

This paper proposes an effective method to estimate travel time distributions from heterogeneous data sources with missing data. The remainder of this paper is organized as follows. Section 2 reviews the literature on the travel time distribution estimation methods. Section 3 briefly introduces Dempster-Shafer (D-S) evidence theory to provide the necessary background. Section 4 presents the proposed heterogeneous data fusion method. Section 5 reports a case study using real-world data from Hong Kong. Section 6 provides conclusions and recommendations for further research.

2. Literature Review

Travel time estimations have been intensively studied for over three decades. Early studies proposed various methods to estimate deterministic travel times, e.g., mean travel time, using a single data source [16–20]. A complete survey of these methods is beyond the scope of this paper; interested readers can refer to comprehensive reviews by Mori et al. and Vlahogianni et al. [8,21].

In the last decade, many research efforts have focused on data fusion techniques to enhance accuracy and robustness of deterministic travel time estimation using multiple data sources. Current data fusion techniques can be broadly classified into statistical, artificial cognition, and probabilistic-based techniques [22]. Statistical based techniques, such as simple convex combination algorithms, use statistical information of data quality to determine weights for different data sources [16,23]. They are relatively widely used due to their simple implementation. However, they are not well suited to fuse different data sources, which are inconsistent and even conflicting. Artificial cognition based techniques combine multiple data sources using artificial intelligence techniques, such as neural networks or genetic algorithms [11]. They can tackle complex data fusion problems, but require large datasets for training, which are generally infeasible for many real-world applications. Probabilistic based techniques typically employ Bayesian and/or D-S evidence theory to provide mathematical reasoning rules for fusing inaccurate and inconsistent data from multiple sources [24–26]. The D-S evidence theory can be regarded as a generalization of Bayesian theory without the requirement of prior knowledge. Nevertheless, most existing studies using D-S evidence theory are restricted to estimation of traffic states (i.e., very congested, congested, medium, smooth or very smooth) rather than precise numerical values of travel times [27–29].

Since no single data source covers the whole network, research efforts have also investigated missing data imputing techniques to enhance data completeness. Missing data imputation can

be broadly classified as prediction and spatial interpolation based techniques. Prediction-based techniques adopt travel time prediction models, such as K-nearest neighbors, kernel regression, and autoregressive integrated moving average, to forecast the missed data from historical data [30–32]. Spatial interpolation techniques impute missing data of a link by using established statistical relationships between the link and its adjacent links [14–16,33]. For all techniques in both categories, incorporation of travel time correlations is well recognized as an effective way to improve imputation performance [32]. However, most missing data imputing techniques assume fixed travel time correlations, which are inadequate to represent the dynamic nature of traffic conditions.

The above studies focused on estimating only deterministic travel times, while ignoring travel time variances. Recent attention has investigated methods to estimate travel time distributions (including means and variances) using a single data source. Dion and Rakha [34] proposed an exponential smoothing filter to estimate travel time distributions using interval detector data. Jenelius and Koutsopoulos [35] developed a maximum likelihood approach to estimate travel time distributions using floating car data. Rahmani et al. [36] used the same data type and proposed a nonparametric approach to estimate travel time distributions. Hans et al. [37] used point detector data and proposed a kinematic wave approach for estimating travel time distributions at signalized intersections. Accurate estimation of travel time distributions is more challenging, because more data with higher quality are required to estimate reliable mean and variance information. Including multiple data sources is obviously beneficial for accurate and robust estimations of travel time distributions, but to the best of our knowledge, few studies have been done for estimating travel time distributions by fusing multiple data sources.

To fill this gap, the current study proposes a heterogeneous data fusion method for estimating travel time distributions, fusing interval and point detector data. In the proposed method, link travel time distributions are first estimated from point detector observations. The spatially missing data issue of point detectors is addressed. Travel time distributions of links without point detectors are imputed based on their spatial correlations with links with point detectors. Estimated link travel time distributions from point detector data are then fused with path travel time distributions obtained from interval detectors. To fuse these two path distributions, a D-S distribution fusion algorithm is proposed, built on D-S evidence theory. An optimization technique is further introduced to update link travel time distributions and their spatial correlations according to the fused travel time distribution.

3. Brief Introduction of the D-S Evidence Theory

The D-S evidence theory was initially developed by Dempster [38], and later extended and refined by Shafer [39]. This theory can be regarded as a generalization of Bayesian inference to tackle uncertainty reasoning based on incomplete information [40,41]. In contrast to Bayesian inference, D-S evidence theory does not assign priori probabilities to unknown propositions or states [42]. Probabilities are assigned only when supporting evidence is available [43]. This provides a flexible framework for decision making by combining cumulative evidence, and has broad applications in many areas, such as expert systems [40,44], artificial intelligence [45,46], false diagnosis [47,48], target recognition [49–51], decision-making [52], information fusion [53], etc.

Let $\Omega = \{S_1, \dots, S_n\}$ be a collectively exhaustive and mutually exclusive set of states, which is also called the frame of discernment. This frame of discernment contains every possible state of a system. A basic probability assignment (BPA) (also called a belief structure) is a function $m : 2^\Omega \rightarrow [0, 1]$, satisfying $m(\phi) = 0$ and $\sum_{A \subset \Omega} m(A) = 1$, where A is a subset of Ω ; and $2^\Omega = \{A | A \in \Omega\}$ is the power set of Ω consisting of all the subsets of Ω . The assigned probability $m(A)$ measures the belief exactly assigned to A . All the assigned probabilities sum to unity and there is no belief in the empty set ϕ . For notational consistency, boldfaced letters represent vectors or matrixes throughout the paper.

Multiple independent evidence can be fused using the traditional Dempster’s combination rule [43–48,51,52]. With BPAs of two independent evidences, m_1 and m_2 , the combination rule is defined as:

$$\begin{aligned}
 m_{fus}(C) &= m_1 \oplus m_2 = \frac{\sum_{A \cap B = C \neq \phi, \forall A, B \subset \Omega} m_1(A) \times m_2(B)}{1 - \eta} \\
 m_{fus}(\phi) &= 0
 \end{aligned}
 \tag{1}$$

where η is the conflict factor, which ranges from 0 to 1 and represents the degree of total conflict between evidences m_1 and m_2 . $1/(1 - \eta)$ is the normalization factor which ensures the sum of BPAs can be unit. η is given by:

$$\eta = \sum_{A \cap B = \phi, \forall A, B \subset \Omega} m_1(A) \times m_2(B)
 \tag{2}$$

Dempster’s combination rule, Equation (1), provides effective reasoning rules for fusing low and moderate conflict evidences. However, in case of high or complete conflict evidences (i.e., η value approach to 1), traditional D-S evidence theory may lead to unreasonable synthesis results. To reduce the degree of evidence conflict, an effective method is to modify the evidence. A common technique [54,55] is to introduce an unknown state, Θ , into the frame of discernment as $\Omega' = \{\Omega, \Theta\}$, where Θ represents the unknown part of the evidence.

As an alternative, several researchers argued that high conflict are mainly caused by unreliable evidences; and thereby they proposed methods to identify and correct the unreliable evidences before the combination [48,51,56]. Overall, the D-S evidence theory provides mathematical reasoning rules for fusing inaccurate and incomplete data from multiple sources. In Section 4.2.2, the D-S evidence theory is employed to fuse travel time distributions from different data sources, which may be high conflict or even complete conflict.

4. Travel Time Distributions Estimated by Fusing Heterogeneous Data Sources

4.1. Problem Statement

Let $G = (N, E)$ be a directed network consisting of a set of nodes, N , and a set of links, E . A link $a_{ij} \in E$ is defined to be the road section between two adjacent nodes with $n_i \in N$ and $n_j \in N$. Travel time of the link is a random variable, T_{ij} , with mean and standard deviation (STD) t_{ij} and σ_{ij} , respectively. The vector of mean travel times for all links is $\mathbf{t} = [\dots, t_{ij}, \dots]^T$, and the variance-covariance matrix between all links is \mathbf{K} . The matrix \mathbf{K} is the variance-covariance matrix of link travel times. In the variance-covariance matrix \mathbf{K} , elements along the diagonal are the variance of link travel times, and off-diagonal elements are the travel time covariance between two links.

Let p^{od} be a path between starting and ending nodes, n_o and n_d , respectively, consisting of λ consecutive links. Let x_{ij}^{od} be a link path incidence variable, where $x_{ij}^{od} = 1$ means that link a_{ij} is on p^{od} and $x_{ij}^{od} = 0$ otherwise. Let $\mathbf{X} = [\dots, x_{ij}^{od}, \dots]^T$ be the vector of link path incidence variables. The path travel time distribution, T^{od} , can be calculated by summing link travel times along the path,

$$T^{od} = \sum_{\forall a_{ij} \in E} T_{ij} x_{ij}^{od}
 \tag{3}$$

Let t^{od} and σ^{od} be the mean and variance of the path travel time distribution, respectively, then:

$$t^{od} = \mathbf{X}^T \mathbf{t}
 \tag{4}$$

$$\sigma^{od} = \sqrt{\mathbf{X}^T \mathbf{K} \mathbf{X}}
 \tag{5}$$

To obtain travel time distribution information, many detectors of different types may be deployed in the network, as shown in Figure 1 for a simple network with interval and point detectors. A pair of interval detectors, e.g., AVI devices, are installed at n_o and n_d of p^{od} to record the set of vehicles

passing them. The path travel time of each recorded vehicle can be obtained by the time difference from entering to leaving the path, and path travel time distribution can be directly estimated from this data, denoted as T_{int}^{pd} . However, the detailed travel time distributions of all links along the path are unknown and the interval detector data covers only a portion of vehicles with relatively poor temporal sampling.

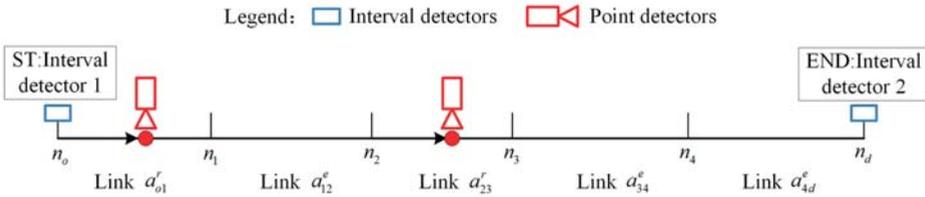


Figure 1. Illustrative example of the heterogeneous data fusion problem.

Point detectors, e.g., loop detectors, are generally deployed for a subset of network links in real applications, e.g., a_{01}^e and a_{23}^e in Figure 1, whereas other links, e.g., a_{12}^e , a_{34}^e and a_{4d}^e , are without detectors. Thus, only travel time distributions of links with point detectors, e.g., T_{01}^e and T_{23}^e , can be directly estimated, while travel time distributions of links without point detectors are unknown, e.g., T_{12}^e , T_{34}^e and T_{4d}^e . Nevertheless, the point detector data tend to have good temporal sampling, since these detectors generally can collect the speeds of all vehicles passing through them.

Obviously, interval and point detector data have distinct spatial and temporal characteristics. Fusing heterogeneous data from both interval and point detectors could be beneficial for estimating travel time distributions for the path and all links either with or without point detectors.

4.2. Proposed Heterogeneous Data Fusion Method

This section presents the proposed heterogeneous data fusion method to estimate travel time distributions for the path and all links either with or without point detectors. Empirical studies have found that travel times can be well represented by either normal, gamma, or lognormal distributions [10,39]. Therefore, to simplify the problem and present the essential concept, it is assumed that all link and path travel time distributions follow the normal distribution [57,58]. Using this normality assumption, the proposed method is to estimate the mean and STD of travel time distributions of the path and all links.

Figure 2 shows that the framework of the proposed heterogeneous data fusion method consists of three steps, described in detail in the following sections. The first step, called data preprocessing, is to estimate path travel time distributions from interval and point detector data, respectively. The second step, called distribution fusion, is to fuse the estimated path travel time distributions by using D-S evidence theory. The last step, called posterior update, is to update link travel time distributions and their travel time correlations based on the fused distribution.

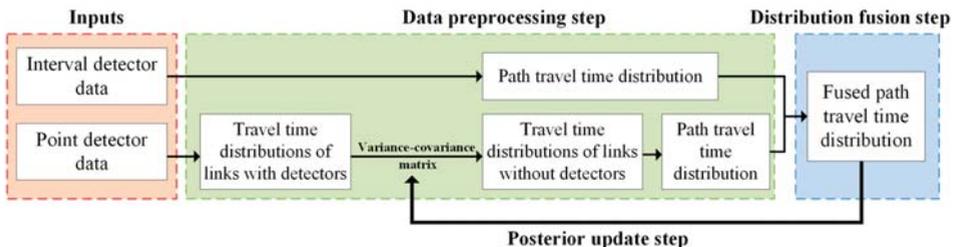


Figure 2. Framework of the proposed heterogeneous data fusion method.

4.2.1. Data Preprocessing Step

This step estimates path travel time distributions from interval and point detector data, independently. The path travel time distribution, T_{int}^{od} , can be directly estimated from interval detector data. Since interval detectors only record a set of vehicles equipped with electronic tags, the limited sample size becomes a critical issue in the estimation, especially for low market penetration applications. Outlier observations can also significantly affect path travel time distribution accuracy, e.g., some vehicles may make stops or detours along the path, leading to atypical travel time observations. To estimate path travel time distribution from interval detector data, the data filtering algorithm proposed by Dion and Rakha [34] was adopted in this study. This data filtering algorithm utilizes a series of low pass filters to remove outlier observations outside a dynamically varying validity window. Such algorithms can perform well in both stable and unstable traffic conditions at low levels of market penetration; and have been successfully applied in the real-time traveler information system (RTIS) in Hong Kong [14]. Thus, an accurate and robust estimation of mean, t_{int}^{od} and STD, σ_{int}^{od} of path travel time distribution can be obtained from interval detector data.

As discussed above, path travel time distribution cannot be directly estimated through point detector data, because only a few links are deployed with point detectors. To estimate the path travel time distributions, links are divided into links with and without point detectors, so that the vector of mean travel time comprises two parts $\mathbf{t}_{poi} = [\mathbf{t}_{poi}^r, \mathbf{t}_{poi}^e]^T$, where \mathbf{t}_{poi}^r and \mathbf{t}_{poi}^e are mean travel times for links with and without point detectors, respectively, at time interval ℓ . The variance-covariance matrix can be divided into four sub-matrixes, $\mathbf{K}_{poi} = \begin{bmatrix} \mathbf{K}_{poi}^{rr} & \mathbf{K}_{poi}^{er} \\ \mathbf{K}_{poi}^{re} & \mathbf{K}_{poi}^{ee} \end{bmatrix}$, where \mathbf{K}_{poi}^{rr} is the variance-covariance matrix for links with point detectors; \mathbf{K}_{poi}^{ee} is the variance-covariance matrix for links without point detectors; \mathbf{K}_{poi}^{er} is the covariance matrix between links without and with point detectors; and $\mathbf{K}_{poi}^{re} = (\mathbf{K}_{poi}^{er})^T$ is the covariance matrix between links with and without point detectors. Let \mathbf{v}_{poi}^r and \mathbf{v}_{poi}^e be vectors of travel time variances for links with and without point detectors, respectively. They are elements along the diagonal of \mathbf{K}_{poi}^{rr} and \mathbf{K}_{poi}^{ee} , respectively.

For a link a_i^j with a point detector, its mean, t_i^j , and STD, σ_i^j , of the link travel time distribution can be obtained from the collected data at the current time interval ℓ , i.e., \mathbf{t}_{poi}^r and \mathbf{K}_{poi}^{rr} can be determined from the point detector data. However, mean travel times for links without point detectors, \mathbf{t}_{poi}^e , should be indirectly estimated. Following Tam and Lam [14], \mathbf{t}_{poi}^e is estimated using spatial correlations between links with and without point detectors:

$$\mathbf{t}_{poi}^e = \mathbf{t}_{upd}^{e,\ell-1} + \mathbf{K}_{upd}^{er,\ell-1} (\mathbf{K}_{poi}^{rr})^{-1} (\mathbf{t}_{poi}^r - \mathbf{t}_{upd}^{r,\ell-1}) \tag{6}$$

where $\mathbf{t}_{upd}^{r,\ell-1}$ and $\mathbf{t}_{upd}^{e,\ell-1}$ are mean travel times for the links with and without point detectors estimated at the previous time interval $\ell - 1$, respectively; $\mathbf{K}_{upd}^{er,\ell-1}$ is the covariance matrix between links without and with point detectors estimated at the previous time interval $\ell - 1$; and $(\mathbf{K}_{poi}^{rr})^{-1}$ is the inverse of \mathbf{K}_{poi}^{rr} .

Similar to Equation (6), \mathbf{v}_{poi}^e in this study was also indirectly estimated using the spatial correlations between links with and without point detectors:

$$\mathbf{v}_{poi}^e = \mathbf{v}_{upd}^{e,\ell-1} + \mathbf{K}_{upd}^{er,\ell-1} (\mathbf{K}_{poi}^{rr})^{-1} (\mathbf{v}_{poi}^r - \mathbf{v}_{upd}^{r,\ell-1}) \tag{7}$$

where $\mathbf{v}_{upd}^{r,\ell-1}$ and $\mathbf{v}_{upd}^{e,\ell-1}$ are travel time variances of the links with and without point detectors at the previous time interval $\ell - 1$, respectively. Therefore, elements along the diagonal of \mathbf{K}_{poi}^{ee} and all elements of \mathbf{K}_{poi}^{rr} are estimated in the current time interval ℓ . It is assumed that $(k_{poi}^{ee})_{ij} = (k_{upd}^{ee,\ell-1})_{ij}, \forall i \neq j$ and $\mathbf{K}_{poi}^{er} = \mathbf{K}_{upd}^{er,\ell-1}$, which means that off-diagonal elements of \mathbf{K}_{poi}^{ee} and all elements of \mathbf{K}_{poi}^{er} are the same as corresponding elements at the previous interval, $\ell - 1$. These two matrixes, \mathbf{K}_{poi}^{ee} and \mathbf{K}_{poi}^{er} , will be updated in the posterior update step in Section 4.2.3.

After $\mathbf{t}_{poi} = [\mathbf{t}_{poi}^r, \mathbf{t}_{poi}^e]^T$ and $\mathbf{K}_{poi} = \begin{bmatrix} \mathbf{K}_{poi}^{rr} & \mathbf{K}_{poi}^{er} \\ \mathbf{K}_{poi}^{re} & \mathbf{K}_{poi}^{ee} \end{bmatrix}$ are determined, the mean, t_{poi}^{od} , and STD, σ_{poi}^{od} , of the path travel time distribution, can be calculated. The vector of link path incidence variables is divided into two groups as $\mathbf{X} = [\mathbf{X}_{poi}^r, \mathbf{X}_{poi}^e]^T$, where \mathbf{X}_{poi}^r and \mathbf{X}_{poi}^e are link path incidence variables for links with and without point detectors, respectively. Then, Equations (4)–(7) for calculating t_{poi}^{od} and σ_{poi}^{od} can be expressed as:

$$t_{poi}^{od} = (\mathbf{X}_{poi}^r)^T \mathbf{t}_{poi}^r + (\mathbf{X}_{poi}^e)^T \mathbf{t}_{poi}^e \tag{8}$$

$$\sigma_{poi}^{od} = \sqrt{(\mathbf{X}_{poi}^r)^T \mathbf{K}_{poi}^{rr} \mathbf{X}_{poi}^r + (\mathbf{X}_{poi}^e)^T \mathbf{K}_{poi}^{ee} \mathbf{X}_{poi}^e + 2(\mathbf{X}_{poi}^e)^T \mathbf{K}_{poi}^{er} \mathbf{X}_{poi}^r} \tag{9}$$

Substituting Equation (6) into Equation (8), the mean travel time can be expressed as:

$$t_{poi}^{od} = (\mathbf{X}_{poi}^r)^T \mathbf{t}_{poi}^r + (\mathbf{X}_{poi}^e)^T \mathbf{t}_{upd}^{e,\ell-1} + (\mathbf{X}_{poi}^e)^T \mathbf{K}_{upd}^{er,\ell-1} (\mathbf{K}_{poi}^{rr})^{-1} (\mathbf{t}_{poi}^r - \mathbf{t}_{upd}^{r,\ell-1}) \tag{10}$$

Therefore, the path travel time distribution, T_{poi}^{od} , can be determined from point detector data.

4.2.2. Distribution Fusion Step

This step fuses two path travel time distributions, T_{int}^{od} and T_{poi}^{od} , estimated from interval and point detectors, respectively. A fusion algorithm is proposed built on the D-S evidence theory. In this study, the frame of discernment, Ω , is defined as a set of mutually exclusive travel time ranges, $\{S_1, \dots, S_i, \dots, S_n\}$, where each travel time range, $S_i = [l_i, u_i]$, is defined by a lower bound l_i and upper bound u_i . The mean travel time for range S_i can be expressed as:

$$E(S_i) = \frac{u_i + l_i}{2} \tag{11}$$

Path travel time distributions estimated by interval and point detectors can be regarded as two independent sets of evidence. Based on the defined travel time ranges, these two path travel time distributions are discretized to obtain corresponding discrete distributions, i.e., histograms, as illustrated in Figure 3a. The resultant discrete distributions, m_{int} and m_{poi} , are respectively modelled as BPAs for T_{int}^{od} and T_{poi}^{od} . Then, $m_*(S_i)$ (either $m_{int}(S_i)$ or $m_{poi}(S_i)$) represents the corresponding probability of travel time range S_i , and can be expressed as:

$$m_*(S_i) = \int_{l_i}^{u_i} f_*(x) dx \tag{12}$$

where $f_*(x)$ is the probability density function of T_*^{od} (either T_{int}^{od} or T_{poi}^{od}). When path travel time distributions follow normal distributions, $m_*(S_i)$ can be expressed as:

$$m_*(S_i) = \Phi_{snd}\left(\frac{u_i - t_*^{rs}}{\sigma_*^{rs}}\right) - \Phi_{snd}\left(\frac{l_i - t_*^{rs}}{\sigma_*^{rs}}\right) \tag{13}$$

where $\Phi_{snd}(\cdot)$ represents the cumulative distribution function (CDF) of the standard normal distribution. In the literature, Hart’s formula [59] is a good numerical approximation approach to calculate $\Phi_{snd}(\cdot)$:

$$\Phi_{snd}(x) \cong \frac{1}{2} - \frac{1}{x\sqrt{2\pi}} \left\{ e^{-x^2/2} - \left[\frac{\pi x^2}{2} + \frac{(1 + 0.282455x^2)^{1/2}}{1 + 0.212024x^2} e^{-x^2/2} \right] \right\} \tag{14}$$

Clearly, BPAs, m_* , satisfies $m_*(\phi) = 0$ and $\sum_{\forall S_i \subset \Omega} m_*(S_i) = 1$.

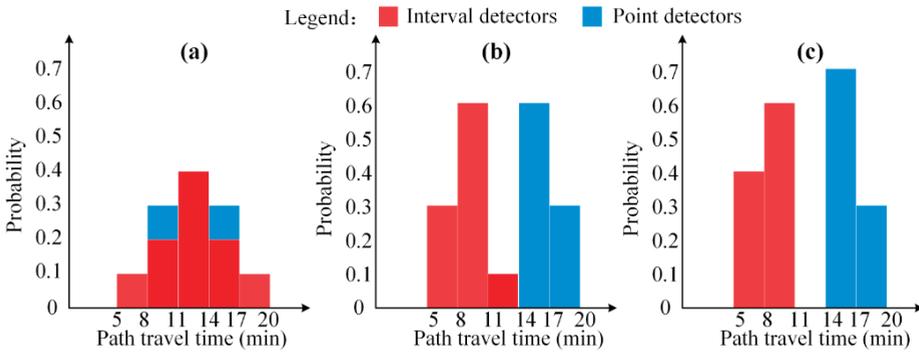


Figure 3. Typical information conflict situations of interval and point detectors: (a) low conflict, (b) high conflict, (c) complete conflict.

Figure 3 illustrates three typical situations of evidence conflict, representing the relationships between interval detector and point detector. Two path travel time are discretized into five travel time ranges as (5, 8), (8, 11), (11, 14), (14, 17) and (17, 20) which constitute the frame of discernment $\Omega = \{S_1, S_2, S_3, S_4, S_5\}$. The corresponding BPAs of two path travel time distributions are shown in Table 1. Figure 3a shows Case 1 that the two evidences have high belief level and low conflict degree, with a large portion of histogram coverage. Figure 3b shows Case 2 that two evidences have low belief level and high conflict degree, with only a small portion of histogram coverage. Figure 3c shows Case 3 that the two evidences are completely conflicted without histogram coverage.

Table 1. Simple example of distribution fusion using Dempster’s combination rule.

Travel Time Ranges	Case 1			Case 2			Case 3		
	$m_{int}(\cdot)$	$m_{poi}(\cdot)$	$m_{fus}(\cdot)$	$m_{int}(\cdot)$	$m_{poi}(\cdot)$	$m_{fus}(\cdot)$	$m_{int}(\cdot)$	$m_{poi}(\cdot)$	$m_{fus}(\cdot)$
S_1	0.1	0	0	0.3	0	0	0.4	0	-
S_2	0.2	0.3	0.2143	0.6	0	0	0.6	0	-
S_3	0.4	0.4	0.5714	0.1	0.1	1	0	0	-
S_4	0.2	0.3	0.2143	0	0.6	0	0	0.7	-
S_5	0.1	0	0	0	0.3	0	0	0.3	-

Table 1 shows the results of evidence fusion by using Dempster’s combination rule, Equation (1). As shown, this combination rule can provide a good estimation of path travel time distribution for Case 1 with a low conflict factor, $\eta = 1 - \sum_{i=1}^5 m_{int}(S_i) \times m_{poi}(S_i) = 1 - (0.2 \times 0.3 + 0.4 \times 0.4 + 0.2 \times 0.3) = 0.72$. The fused BPA is calculated from $m_{fus}(S_i) = m_{int}(S_i) \times m_{poi}(S_i) / (1 - \eta)$ (e.g., $m_{fus}(S_3) = 0.4 \times 0.4 / (1 - 0.72) = 0.5714$). After distribution fusion, travel time ranges, S_2, S_3 and S_4 , supported by both evidence sets, are strengthened in a reasonable way.

However, for Cases 2 with high conflict factor, $\eta = 1 - (0.1 \times 0.1) = 0.99$, the Dempster’s combination rule can lead to an incorrect fusion result, $m_{fus}(S_3) = (0.1 \times 0.1) / (1 - 0.99) = 1$, given both evidence sets afford little support to S_3 . This situation is known as Zadeh’s paradox in the literature. Further, Dempster’s combination rule cannot be used for Case 3 of the completely conflict situation. In this case, $m_{int}(\cdot)$ and m_{poi} cannot be fused, because $\eta = 1$ so all $m_{fus}(S_i)$ become infinite.

To reduce the degree of data conflict, the generalized combination rule, Equation (2), is adopted in this study, by introducing the unknown state into the frame of discernment, $\Omega = \{S_1, \dots, S_i, \dots, S_n, \Theta\}$. Subsequently, to construct BPA m_* (either m_{int} or m_{poi}), a pre-defined small probability $\alpha_\Theta = m_*(\Theta)$, (e.g., $\alpha_\Theta = 0.05$), is set for the unknown state Θ . Then, the path travel time distribution between

$t_*^{od} + \Phi_{snd}^{-1}(\alpha_{\Theta}/2)\sigma_*^{od}$ and $t_*^{od} + \Phi_{snd}^{-1}(1 - \alpha_{\Theta}/2)\sigma_*^{od}$ is discretized to obtain $m_*(S_i)$, where $\Phi_{snd}^{-1}(\cdot)$ is the inverse CDF of the standard normal distribution (e.g., $\Phi_{snd}^{-1}(0.025) = -1.96$ and $\Phi_{snd}^{-1}(0.975) = 1.96$).

High and complete conflict situations are usually due to various data quality from the different detectors. To differentiate data sources with varying quality, an information quality parameter [48] is adopted in this study to assign higher weighting to data sources with better information quality. Let w_{int} and w_{poi} be the information quality weights for the path travel time distribution from interval and point detectors respectively. In this study, w_{int} and w_{poi} are expressed as a function of sample size and travel time variance:

$$w_{int} = \frac{1 - (1 - \beta_{int})^{N_{int}}}{(\sigma_{int}^{od})^2} \tag{15}$$

$$w_{poi} = \frac{1 - (1 - \beta_{poi})^{N_{poi}}}{(\sigma_{poi}^{od})^2} \tag{16}$$

where N_{int} is the sample size collected by interval detectors; N_{poi} is the average sample size for all point detectors along the path; β_{int} and β_{poi} are sensitivity parameters for interval and point detectors, respectively, which should be calibrated independently. Other types of information quality function could also be used in practice.

Applying different weightings w_{int} and w_{poi} , the BPA m_* (either m_{int} or m_{poi}) is adjusted using following formula [48]:

$$\bar{m}_* = \begin{cases} \bar{m}_*(S_i) = \frac{w_*}{w_{max}} \cdot m_*(S_i), \forall S_i \subseteq \Omega \\ \bar{m}_*(\Theta) = 1 - \sum_{i=1}^n \bar{m}_*(S_i) \end{cases} \tag{17}$$

where $w_{max} = \max(w_{int}, w_{poi})$ is the larger between w_{int} and w_{poi} . Substituting the adjusted BPAs into Equations (1) and (2), the fused BPA, m_{fus} , can be determined following the generalized combination rule:

$$m_{fus} = \begin{cases} m_{fus}(S_i) = \frac{\bar{m}_{int}(S_i) \times \bar{m}_{poi}(S_i) + \bar{m}_{int}(S_i) \times \bar{m}_{poi}(\Theta) + \bar{m}_{int}(\Theta) \times \bar{m}_{poi}(S_i)}{1 - \eta}, \forall S_i \subseteq \Omega \\ m(\Theta) = \frac{\bar{m}_{int}(\Theta) \times \bar{m}_{poi}(\Theta)}{1 - \eta} \\ \eta = 1 - \bar{m}_{int}(\Theta) \times \bar{m}_{poi}(\Theta) - \sum_{i=1}^n [\bar{m}_{int}(S_i) \times \bar{m}_{poi}(S_i) + \bar{m}_{int}(\Theta) \times \bar{m}_{poi}(S_i) + \bar{m}_{poi}(\Theta) \times \bar{m}_{int}(S_i)] \end{cases} \tag{18}$$

Table 2 illustrates the distribution fusion built on the generalized combination rule using the same example as in Table 1. In this example, $m_{int}(\Theta) = m_{poi}(\Theta) = 0.05$ are set; and two BPAs, m_{int} and m_{poi} are modified to reflect this setting. Information quality parameters $w_{int} = 0.8$ and $w_{poi} = 0.6$ are used for interval and point detectors, respectively. All BPAs, m_{poi} , for these cases are adjusted to $\bar{m}_{poi}(S_i) = m_{poi}(S_i) \times 0.6/0.8, \forall S_i \subseteq \Omega$; and $\bar{m}_{poi}(\Theta) = 1 - \sum_{i=1}^5 \bar{m}_{poi}(S_i) = 1 - 0.95 \times 0.6/0.8 = 0.288$. The generalized combination rule, Equation (16), was adopted for fusing path travel time distribution.

Table 2. Simple example of distribution fusion using the generalized combination rule.

Travel Time Ranges	Case 1			Case 2			Case 3		
	$m_{int}(\cdot)$	$m_{poi}(\cdot)$	$m_{fus}(\cdot)$	$m_{int}(\cdot)$	$m_{poi}(\cdot)$	$m_{fus}(\cdot)$	$m_{int}(\cdot)$	$m_{poi}(\cdot)$	$m_{fus}(\cdot)$
S_1	0.075	0	0.0410	0.275	0	0.2415	0.375	0	0.3337
S_2	0.2	0.275	0.2075	0.6	0	0.5270	0.575	0	0.5116
S_3	0.4	0.4	0.4756	0.075	0.075	0.0874	0	0	0.0000
S_4	0.2	0.275	0.2075	0	0.6	0.0687	0	0.675	0.0783
S_5	0.075	0	0.0410	0	0.275	0.0315	0	0.275	0.0319
Θ	0.05	0.05	0.0273	0.05	0.05	0.0439	0.05	0.05	0.0445

Table 2 shows that the generalized combination rule provides a reasonable outcome for Case 1 (i.e., low conflict situation). More importantly, this generalized combination rule can well address the distribution fuse problem for Case 2 (i.e., the high conflict situation). Introducing Θ significantly reduced the conflict factor η to 0.6614. The probability of S_3 , which has little support from both evidence sets, is only slightly strengthened as $m_{fus}(S_3) = (0.075 \times 0.056 + 0.075 \times 0.288 + 0.05 \times 0.056) / (1 - 0.6614) = 0.0874$. The probabilities of other travel time ranges, S_1 , S_2 , S_4 , and S_5 , are reduced, but a higher weighting is given to the data source with better data quality (i.e. m_{int}). The generalized combination rule also addressed the distribution fusion for Case 3 (i.e., complete conflict situation), which cannot be fused using Dempster’s combination rule.

From the fused BPA, m_{fus} , the corresponding mean and STD, can be expressed as:

$$\theta = \frac{1}{1 - m_{fus}(\Theta)} \tag{19}$$

$$t_{fus}^{od} = \sum_{i=1}^n \theta \cdot m_{fus} \cdot (S_i) \cdot E(S_i) \tag{20}$$

$$\sigma_{fus}^{od} = \sqrt{\sum_{i=1}^n (E(S_i) - t_{fus}^{od})^2 \cdot \theta \cdot m_{fus} \cdot (S_i)} \tag{21}$$

where θ is the adjustment parameter to assign the probability of the unknown state to each travel time range. Thus, the proposed D-S distribution fusion algorithm can estimate path travel time distributions by fusing two path travel time distributions from interval and point detector data, even in the cases of extreme conflict between the data sets.

4.2.3. Posterior Update Step

This step updates the link travel time distributions and their spatial correlations based on the fused path travel time distribution. An optimization technique is proposed to update the travel time means (i.e., $t_{poi} = [t_{poi}^r, t_{poi}^e]^T$) and variance-covariance matrix (i.e., $K_{poi} = \begin{bmatrix} K_{poi}^{rr} & K_{poi}^{re} \\ K_{poi}^{re} & K_{poi}^{ee} \end{bmatrix}$) estimated in the data preprocessing step.

Let $t_{upd} = [t_{upd}^r, t_{upd}^e]^T$ and $K_{upd} = \begin{bmatrix} K_{upd}^{rr} & K_{upd}^{re} \\ K_{upd}^{re} & K_{upd}^{ee} \end{bmatrix}$ be the updated travel time means and covariance matrix, respectively where $(k_{upd}^{ee})_{ij}$ is the element at row i and column j of K_{upd}^{ee} . This study uses $t_{upd}^r = t_{poi}^r$ and $K_{upd}^{rr} = K_{poi}^{rr}$, because t_{poi}^r and K_{poi}^{rr} are directly obtained from point detector data and assumed to be accurate. Therefore, to update the link travel time covariance matrix, only K_{upd}^{ee} and K_{upd}^{re} sub-matrixes need to be updated, since $K_{upd}^{re} = (K_{upd}^{ee})^T$ holds. Accordingly, the optimization problem of updating the spatial correlations is formulated as the following nonlinear programming problem:

$$M1 \min \left(\sum_{\forall i} \sum_{\forall j} \left((k_{upd}^{ee})_{ij} - (k_{poi}^{ee})_{ij} \right)^2 + \sum_{\forall i} \sum_{\forall j} 2 \left((k_{upd}^{er})_{ij} - (k_{poi}^{er})_{ij} \right)^2 \right) \tag{22}$$

Subject to:

$$t_{fus}^{od} = (X_{poi}^r)^T t_{upd}^r + (X_{poi}^e)^T t_{upd}^{e,\ell-1} + (X_{poi}^e)^T K_{upd}^{er} (K_{poi}^{rr})^{-1} (t_{poi}^r - t_{upd}^{r,\ell-1}) \tag{23}$$

$$(\sigma_{fus}^{od})^2 = (X_{poi}^r)^T K_{poi}^{rr} X_{poi}^r + (X_{poi}^e)^T K_{upd}^{ee} X_{poi}^e + 2(X_{poi}^e)^T K_{upd}^{er} X_{poi}^r \tag{24}$$

The nonlinear programming M1 has a convex objective function and two linear constraints. To ensure K_{upd} is stable over time, objective function (22) minimizes the total difference of

updating elements in both $\mathbf{K}_{\text{upd}}^{\text{ee}}$ and $\mathbf{K}_{\text{upd}}^{\text{er}}$ sub-matrixes. Constraints (23) and (24), derived from Equations (9) and (10), ensure that the summation of means and variances of corresponding link travel time distributions are equal to that of the fused path travel time distribution, i.e., $t_{\text{fus}}^{\text{od}}$ and $(\sigma_{\text{fus}}^{\text{od}})^2$. This M1 problem is a typical quadratic programming problem. A unique solution can be determined using several efficient algorithms, such as the quadprog function in MatLab.

Once \mathbf{K}_{upd} is determined, the vector of travel time means for links without point detectors, $\mathbf{t}_{\text{upd}}^{\text{e}}$ are updated as:

$$\mathbf{t}_{\text{upd}}^{\text{e}} = \mathbf{t}_{\text{upd}}^{\text{e},\ell-1} + \mathbf{K}_{\text{upd}}^{\text{er}} (\mathbf{K}_{\text{poi}}^{\text{rr}})^{-1} (\mathbf{t}_{\text{poi}}^{\text{r}} - \mathbf{t}_{\text{upd}}^{\text{r},\ell-1}) \quad (25)$$

The updated \mathbf{t}_{upd} and \mathbf{K}_{upd} are used for estimating travel time distributions of links without point detectors in the subsequent time interval. The detailed steps of the Algorithm 1 are summarized as follows.

Algorithm 1

Step 1. Data preprocessing stage:

Estimate $T_{\text{int}}^{\text{od}}$ from interval detector data at current interval ℓ .

Estimate $\mathbf{t}_{\text{poi}}^{\text{r}}$ and $\mathbf{K}_{\text{poi}}^{\text{rr}}$ for links with point detectors at current interval ℓ .

Deduce $\mathbf{t}_{\text{poi}}^{\text{e}}$ and $\mathbf{v}_{\text{poi}}^{\text{e}}$ for links without point detectors using Equations (6) and (7).

Estimate $T_{\text{poi}}^{\text{od}}$ using Equations (9) and (10).

Step 2. Distribution fusion stage:

Estimate $T_{\text{fus}}^{\text{od}}$ by fusing $T_{\text{int}}^{\text{od}}$ and based on Equations (11)–(21).

Step 3. Posterior update stage:

Update \mathbf{K}_{upd} using Equations (22)–(24); and update \mathbf{t}_{upd} using Equation (25).

Set $\mathbf{K}_{\text{upd}}^{\ell-1} = \mathbf{K}_{\text{upd}}$, and $\mathbf{t}_{\text{upd}}^{\ell-1} = \mathbf{t}_{\text{upd}}$.

Go to Step 1 for next time interval.

5. Numerical Experiments

Performance of the proposed heterogeneous data fusion method was investigated using real-world data from Hong Kong, as shown in Figure 4. A path from Aberdeen tunnel in Hong Kong Island to the Cross Harbor tunnel (CHT) in Kowloon urban area was selected for this case study. CHT is the most congested of the three tunnels connecting Kowloon urban area and Hong Kong Island. The total travel distance of the chosen path was 3.7 km with free-flow travel time 3.6 min. There were 11 links in the chosen path, with only two, Links 1 and 5, equipped with Autoscope video image detectors (VIDs), which is a popular type of point detector. Two AVI devices were installed at the beginning and end of the chosen path for automatic toll collection. Market penetration of AVI systems was approximately 40%. Real-time AVI data were also utilized for the implementation of RTIS (real-time traveler information systems) in Hong Kong [14]. Detailed information of this AVI system was provided in Tam and Lam [14].

Traffic data from both interval and point detectors were collected during (07:00–23:00) of a typical weekday: Wednesday, 20 August 2014. An offline link travel time covariance matrix was obtained from RTIS [14] as the initial \mathbf{K}_{fus} . To evaluate the performance of the proposed heterogeneous data fusion method, a manual license plate matching survey was performed. Video recording equipment was set at the starting and end nodes of the chosen path to record the license plate readings of vehicles. The vehicles recorded at the starting and end nodes were manually matched. Path travel times of matched vehicles were computed as ground truth for accuracy validation.



Figure 4. Study area location.

5.1. Evaluation Metrics

Two widely accepted metrics, mean absolute percentage error (MAPE) and root mean square error (RMSE), were adopted to evaluate the accuracy of the estimated mean of path travel time distributions:

$$MAPE_t = \frac{100\%}{n} \cdot \sum_{\ell=1}^n \frac{|t_{fus}^{od} - t_{obs}^{od}|}{t_{obs}^{od}} \tag{26}$$

$$RMSE_t = \sqrt{\frac{1}{n} \cdot \sum_{\ell=1}^n (t_{fus}^{od} - t_{obs}^{od})^2} \tag{27}$$

where n is the number of time intervals during the period of interest, and t_{obs}^{od} is the ground truth observed mean travel time obtained from the field survey at time interval ℓ . Smaller $MAPE_t$ and $RMSE_t$ indicate higher accuracy of the estimated mean travel time.

The MAPE and RMSE concepts were extended to evaluate the accuracy of the estimate STD of the path travel time as:

$$MAPE_{\sigma} = \frac{100\%}{n} \cdot \sum_{\ell=1}^n \frac{|\sigma_{fus}^{od} - \sigma_{obs}^{od}|}{\sigma_{obs}^{od}} \tag{28}$$

$$RMSE_{\sigma} = \sqrt{\frac{1}{n} \cdot \sum_{\ell=1}^n (\sigma_{fus}^{od} - \sigma_{obs}^{od})^2} \tag{29}$$

where σ_{obs}^{od} represents the ground truth observed travel time STD obtained from the field survey at time interval ℓ .

For many transportation applications, it is meaningful to construct a travel time interval at a given confidence level from the estimated travel time distribution [60,61]. The travel time interval accuracy represents the integrated accuracy of both the estimated mean and STD. Two metrics were adopted to evaluate these accuracies: probability outside the predicted (estimated) time interval (POPI) and probability outside the observed time interval (POOI) [62]. The POPI measures the percentage of observed data outside the estimated travel time interval, while the POOI measures the percentage of estimated distribution outside the observed travel time interval.

Let $l_{fus} = \Phi_{fus}^{-1}(\alpha/2)$ and $u_{fus} = \Phi_{fus}^{-1}(1 - \alpha/2)$ be the lower and upper bounds of the estimated travel time interval, respectively, at confidence level $1 - \alpha$, where $\Phi_{fus}^{-1}(\cdot)$ is the inverse CDF of the estimated path travel time distribution. Then:

$$POPI = \frac{100\%}{n} \cdot \sum_{\ell=1}^n \left(1 - \frac{\Phi_{obs}(u_{fus}) - \Phi_{obs}(l_{fus})}{1 - \alpha} \right) \quad (30)$$

where $\Phi_{obs}(\cdot)$ is the CDF of the observed travel time distributions. The POPI value ranges from 0 to 1. The smaller POPI indicates capture of larger proportion of observed data, i.e., higher accuracy of the estimated travel time interval. As noted by Shi et al. [62], this POPI metric is very useful, but tends to exhibit bias for situations of wide travel time intervals due to large STD errors.

As an alternative, POOI metric is the percentage of estimated distribution outside the observed travel time interval. Let $l_{obs} = \Phi_{obs}^{-1}(\alpha/2)$ and $u_{obs} = \Phi_{obs}^{-1}(1 - \alpha/2)$ denote the lower and upper bounds of the observed travel time interval, respectively, at confidence level $1 - \alpha$, where $\Phi_{obs}^{-1}(\cdot)$ is the inverse CDF of the observed path travel time distribution, and $\Phi_{fus}(\cdot)$ denotes the CDF of the estimated travel time distribution. Then:

$$POOI = \frac{100\%}{n} \cdot \sum_{\ell=1}^n \left(1 - \frac{\Phi_{fus}(u_{obs}) - \Phi_{fus}(l_{obs})}{1 - \alpha} \right) \quad (31)$$

POOI also ranges [0, 1], and larger POOI indicates lower estimated travel time interval accuracy, because a larger proportion is outside the observed travel time interval. Therefore, the POPI and POOI matrices are complementary to evaluate the estimated path travel time distribution accuracy.

5.2. Experimental Results

This section reports experimental results of the case study using the proposed heterogeneous data fusion method. Travel time distributions for the chosen path and links were estimated every 2 min. The probability of the unknown state for both interval and point detectors was set as $\alpha_{\Theta} = m_{int}(\Theta) = m_{poi}(\Theta) = 0.05$, and sensitivity parameters in Equations (15) and (16) were set as $\beta_{int} = 0.2$ and $\beta_{poi} = 0.8$, according to the sensitive analysis results obtained from Dion and Rakha [34]. Setting $\beta_{poi} = 4\beta_{int}$ assigns a higher level of information quality to the interval detector than point detector data, given the same sample sizes.

Figure 5 shows two path travel time distributions, T_{int}^{od} and T_{poi}^{od} , estimated from interval and point detectors, respectively, in the data preprocessing step. Travel time intervals were constructed for the 95% confidence level, i.e., $\alpha_{\Theta} = 0.05$, for both interval and point detectors, shown in blue and red, respectively. Observed data from the field survey, shown in green dots, were only used for accuracy validation. As shown in the figure, two estimated travel time intervals from different data sources can cover most observed data well during the period of interest. The two estimated travel time distributions show high consistency during off-peak periods (21:00–23:00 and 7:00–7:30), slight inconsistency during inter-peak periods (10:00–16:00), and high inconsistency during peak periods (7:30–10:00 and 16:00–21:00). In general, T_{int}^{od} tended to have higher accuracy than T_{poi}^{od} . This was expected, since T_{int}^{od} was estimated from interval detector data, whereas T_{poi}^{od} was estimated from point detector data through spatial interpolation. Such a result also justified the chosen sensitivity parameters, reflecting the higher level of information quality for the interval detector data.

Figure 6 shows the resultant path travel time distribution after fusing the two path travel time distributions from Figure 5. A confidence level of 80%, i.e., $\alpha = 0.2$, was used to construct the travel time interval and calculate POPI and POOI metrics. The proposed heterogeneous data fusion method provided an accurate and robust estimation of mean travel time, t_{fus}^{od} , throughout the period of interest, with $MAPE_t = 7.1\%$. However, the relative large $MAPE_{\sigma} = 17.9\%$ showed that the proposed method overestimated path travel time distribution STD, σ_{fus}^{od} , for the period of interest. This highlights

the challenge of accurately estimating σ_{fus}^{od} in congested road networks. One major reason may be the difficulty of estimating σ_{obs}^{od} of the population using biased samples with various data quality. Fortunately, the slight STD over estimation could be beneficial to most travelers with risk-averse attitudes regarding travel time uncertainty. $POPI = 15.7\%$, somewhat better than the target (20%), which indicates that a high proportion (84.3%) of observation data was well covered by the estimated path travel time interval. It can also be seen from the figure that the estimated interval was not too wide, given the relative large STD error. $POOI = 25.6\%$, which was somewhat larger than the target (20%). Thus, overall the POPI and POOI metrics verified that the proposed heterogeneous data fusion method could obtain accurate and robust estimations of the path travel time interval (i.e., path travel time distribution) by fusing heterogeneous interval and point detector data.

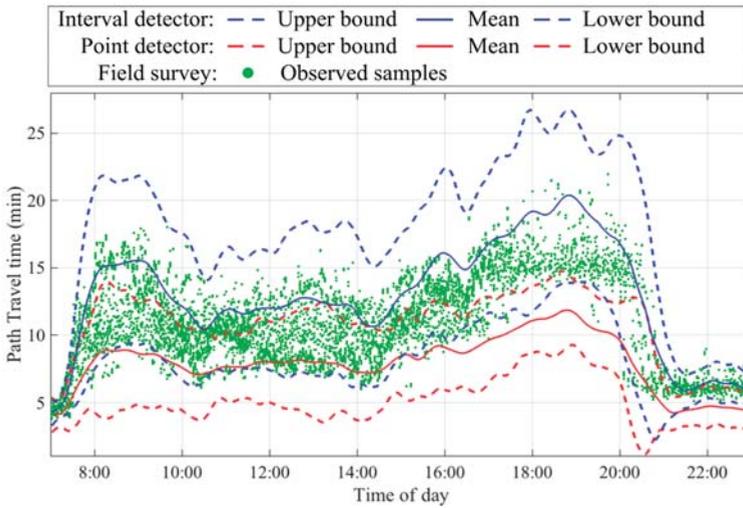


Figure 5. Two path travel time distributions obtained in the data preprocessing step.

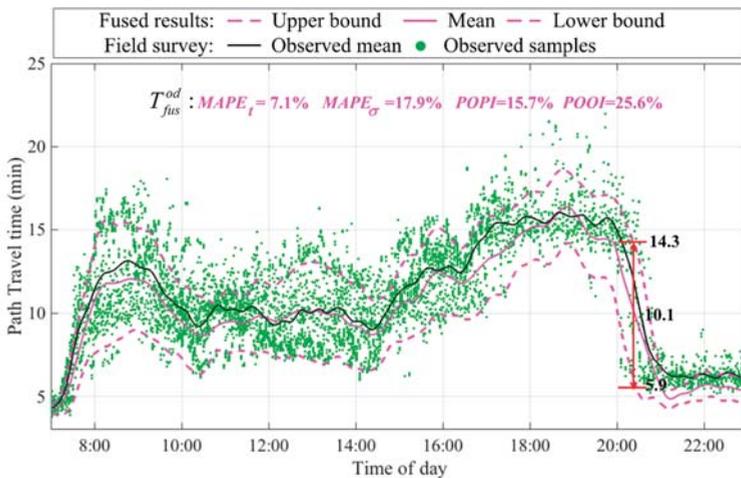


Figure 6. Fused path travel time distribution during the period of interest.

5.3. Comparison of Data Fusion and Single Data Source Results

In this section, the effectiveness of the proposed heterogeneous data fusion method was investigated by comparing data fusion results with those estimated from single data source. The estimated path travel time distribution (i.e., T_{int}^{od}) from single interval detector data was shown in Figure 5 in blue. The estimated path travel time distribution from single point detector data (denoted by \tilde{T}_{poi}^{od}) was shown in Figure 7 in blue, which was different from the T_{poi}^{od} estimation shown in Figure 5. It should be noted that \tilde{T}_{poi}^{od} was generated using fixed offline spatial correlations obtained from RTIS, and T_{poi}^{od} was generated by the proposed heterogeneous data fusion method using the updated spatial correlations.

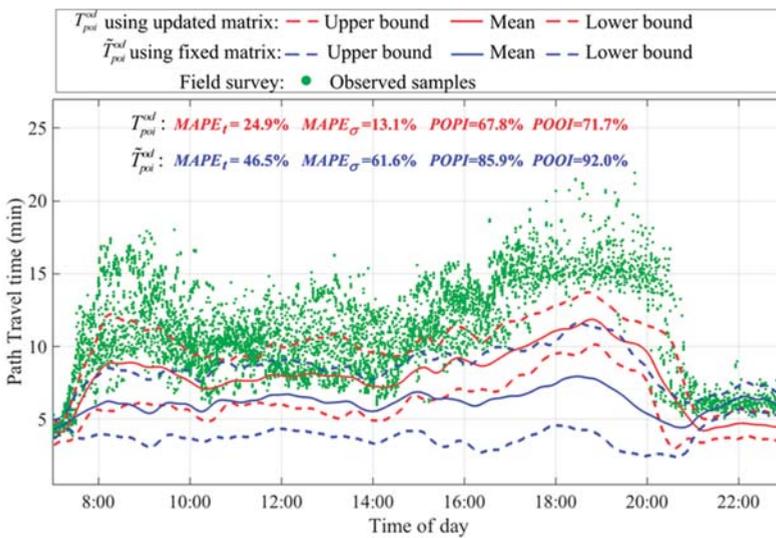


Figure 7. Path travel time distributions estimated from point detector data by using updated and fixed spatial correlations.

Figure 7 shows travel time intervals of \tilde{T}_{poi}^{od} and T_{poi}^{od} in blue and red colors for comparison. The 80% confidence level was used for construing travel time intervals and calculating $POPI$ and $POOI$ metrics. As illustrated, by using updated spatial correlations, the accuracy of the path travel time distribution estimated from point detector data was significantly improved. The $MAPE_t$, $MAPE_\sigma$, $POPI$, and $POOI$ metrics were reduced by 46.4% (i.e., $1 - 24.9\%/46.5\%$), 78.9%, 21.1%, and 22.1%, respectively. This validates the effectiveness of the proposed optimization technique for updating travel time spatial correlations. Such a result also highlights the necessity for considering the dynamic nature of travel time spatial correlations in congested road networks, and implies that current spatial interpolation techniques [14,15] built on fixed spatial correlations may lead to considerable errors when imputing missing data.

Table 3 summarizes the evaluation metrics for all path travel time distributions estimated from point detector, \tilde{T}_{poi}^{od} ; interval detector, T_{int}^{od} ; and fused data, T_{fus}^{od} . Amongst these three distributions, the accuracy of \tilde{T}_{poi}^{od} was the poorest, with $MAPE_t = 46.5\%$ and $MAPE_\sigma = 61.6\%$. The $POPI = 85.9\%$ indicates that a large proportion (85.9%) of observations falling outside the travel time interval of \tilde{T}_{poi}^{od} . The $POOI = 92.0\%$ shows that almost whole travel time range of \tilde{T}_{poi}^{od} was out of the observed time interval. The accuracy of T_{int}^{od} was somewhat superior, with $MAPE_t = 17.1\%$, $MAPE_\sigma = 76.9\%$, $POPI = 26.4\%$ and $POOI = 48.9\%$. As shown, T_{fus}^{od} , using the proposed data fusion method, was the

best for all evaluation metrics. By fusing interval and point detector data, the $MAPE_t$, $MAPE_\sigma$, $POPI$ and $POOI$ metrics were respectively reduced by 58.5% (i.e., 1–7.1%/17.1%), 76.7%, 40.5%, and 47.6%, when compared to that of T_{int}^{od} . Thus, the proposed heterogeneous data fusion method can significantly improve the accuracy of path travel time distribution estimations from interval and point detectors.

Table 3. The accuracy of data fusion results and single data source results.

Data Source	Estimated Mean		Estimated STD		POPI	POOI
	MAPE	RMSE (min)	MAPE	RMSE (min)		
Point detectors	46.5%	2.32	61.6%	0.75	85.9%	92.0%
Interval detectors	17.1%	1.42	76.9%	1.01	26.4%	48.9%
Data fusion	7.1%	0.85	17.9%	0.35	15.7%	25.6%

Fusion of interval and point detector data can improve the accuracy of travel time distributions for links without point detectors. When only point detector data were used, travel time distributions for links without point detectors were indirectly estimated through the fixed spatial correlations. Fusing interval and point detector data provided better estimations of link travel time distributions from the updated spatial correlations. Figure 8 compares individual link travel time distributions estimated from point detector data and the proposed data fusion method. Ground truth data for these link travel time distributions were not available for quantitative analysis of estimation accuracy. Nevertheless, link travel time distributions estimated from the proposed heterogeneous data fusion method better capture dynamic traffic conditions, with more distinct peaks occurring during the morning and evening peak periods. The much superior accuracy of path travel time distribution estimation (see Table 3) also justifies this visual observation, because the path travel time distribution is the summation of corresponding link travel time distributions along the path.

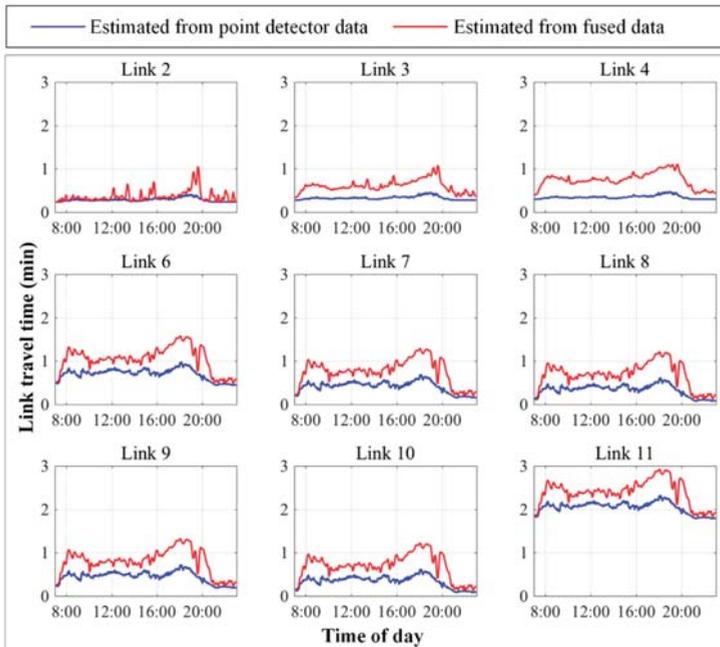


Figure 8. Individual link travel time distribution estimated from point detector data and fused data.

5.4. Comparison of Different Distribution Fusion Algorithms

This section investigates the effectiveness of the proposed D-S distribution fusion algorithm built on the D-S evidence theory. To further evaluate and benchmark the proposed algorithm, a linear combination fusion algorithm built on the linear combination approach was also implemented. The linear combination approach (or simple convex combination approach) has been widely used as a simple and effective technique to fuse two independent estimations of mean travel times [11],

$$t_{fus}^{od} = \frac{w_{int}}{w_{int} + w_{poi}} t_{int}^{od} + \frac{w_{poi}}{w_{int} + w_{poi}} t_{poi}^{od} \tag{32}$$

where w_{int} and w_{poi} are the data quality of interval and point detectors, respectively, as defined in Equations (15) and (16). This study extended the linear combination approach to fuse two independent STD estimations, as:

$$\sigma_{fus}^{od} = \frac{w_{int}}{w_{int} + w_{poi}} \sigma_{int}^{od} + \frac{w_{poi}}{w_{int} + w_{poi}} \sigma_{poi}^{od} \tag{33}$$

Assuming normal distributions, this extended linear combination fusion algorithm can be used to fuse path travel time distributions from interval and point detectors.

In this study, the same set of input data was used to validate the results of the proposed D-S distribution fusion and the linear combination fusion algorithms. Path travel time distributions of interval and point detectors obtained in the data preprocessing step, shown in Figure 5, were employed as the input data. Figure 9 reports the fused path travel time distributions using these two algorithms. As shown, the proposed D-S distribution fusion algorithm produces better of path travel time distribution estimates than the linear combination fusion algorithm. The proposed algorithm can significantly reduce $MAPE_t$, $MAPE_\sigma$, $POPI$, and $POOI$ metrics by 58.6%, 15.3%, 37.2%, and 38.0%, respectively, compared to the linear combination fusion algorithm. This result indicates that the D-S evidence theory is effective for fusing inaccurate and inconsistent distribution data from multiple sources under various information conflict situations, including highly consistent, slightly inconsistent, and highly inconsistent situations.

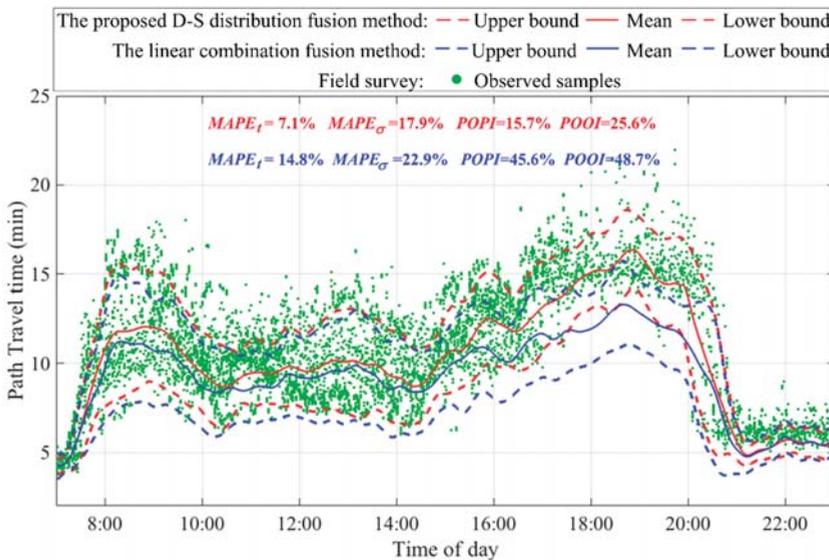


Figure 9. Path travel times of different methods during the period of interest.

6. Conclusions and Future Research

Provision of travel time distribution information is a crucial requirement for travelers to make reliable path choice decisions incorporating travel time uncertainties. With advances in information and communication technologies, interval detectors (such as automatic vehicle identification devices) and point detectors (such as loop detectors) are being increasingly deployed in road networks. These interval and point detectors generate heterogeneous data sources with distinct characteristics of data quality and network coverage. Fusing these heterogeneous data can be beneficial for robust and accurate estimation of travel time distribution information.

This paper proposed a heterogeneous data fusion method to estimate travel time distributions, fusing heterogeneous data from point and interval detectors. The proposed method consisted of three steps. The first step, i.e., data preprocessing, was to respectively estimate path travel time distributions from interval and point detector data. The spatially missing data issue of point detectors was addressed. The travel time distributions of links without point detectors were imputed based on their spatial correlations with links that had point detectors. The second step, i.e., distribution fusion, was to fuse these two path travel time distributions estimated from interval and point detectors. A D-S distribution fusion algorithm built on the Dempster-Shafer evidence theory was proposed to fuse path travel time distributions from different data sources with various information qualities. The third step, i.e., posterior update, was to update link travel time distributions and their spatial correlations. The problem of updating spatial correlations was formulated and solved as a quadratic programming problem with a convex objective function and two linear constraints.

To validate the accuracy of the proposed heterogeneous data fusion method, a case study was performed using real-world data from RTIS in Hong Kong. The results validated that the proposed method can obtain robust and accurate estimations of path travel time distributions in congested road networks. Compared with either interval or point detectors alone, the proposed data fusion method can significantly reduce estimation errors for path travel time distributions with respect to $MAPE_t$, $MAPE_\sigma$, $POPI$, and $POOI$ metrics. The proposed D-S distribution fusion algorithm was also compared to a linear combination algorithm for the same case study, and it showed that the proposed D-S distribution fusion algorithm can generate a robust and accurate fusion of travel time distributions over the whole period of interest, including highly consistent, slightly inconsistent, and highly inconsistent situations for the different data sources. Furthermore, the results of the case study indicated that the proposed optimization technique can effectively update travel time spatial correlations under dynamic traffic conditions, and incorporation of updated spatial correlations greatly enhanced estimation accuracy of travel time distributions of the path and all links without point detectors. Therefore, the proposed D-S distribution algorithm was validated to be effective for fusing travel time distributions from different data sources under various information conflict situations, including highly consistent, slightly inconsistent, and highly inconsistent situations.

There are several worthwhile directions for future research. First, travel times in this study were assumed to follow normal distributions. However, several previous studies have found that travel times in congested road networks could be better represented by asymmetric distributions with strong positive skew, e.g., lognormal, gamma, or Burr distributions [10,57]. The proposed heterogeneous data fusion method can be easily extended to other types of distributions with two parameters, e.g., lognormal or gamma, by replacing Equation (14) with corresponding methods to calculate the cumulative distribution function. Second, the spatial interpolation proposed by Tam and Lam [14] was adopted in this study for imputing the travel time distributions of links without point detectors. However, other effective spatial interpolation techniques have been proposed, such as Kriging [15]. Integrating these alternative spatial interpolation techniques into the proposed heterogeneous data fusion method warrants further study. Third, the proposed data fusion method only considered heterogeneous data from point and interval detectors. How to extend the proposed method to incorporate floating car data needs further investigation. Fourth, the case study only involved a specific path. Extension of the proposed method to fuse travel time distributions of multiple

paths between a pair of nodes is an interesting topic for further investigation. Last but not the least, travel time distributions were estimated in this study for the current time interval. Extension of the proposed data fusion method to the problem of short term travel time distribution prediction is another interesting topic for further study.

Acknowledgments: The work described in this paper was supported by research grants from the National Key Research and Development Program of China (No. 2017YFB0503604), the National Natural Science Foundation of China (Nos. 41231171 and 41571149), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 152628/16E), the Research Institute for Sustainable Urban Development of the Hong Kong Polytechnic University (1-ZVEW), and the Natural Science Foundation of Hubei Province (2016CFB568).

Author Contributions: Chaoyang Shi, Bi Yu Chen, William H. K. Lam and Qingquan Li provided the idea of this study; Chaoyang Shi implemented the proposed method, carried out the case study; Chaoyang Shi and Bi Yu Chen wrote the manuscript; and William H. K. Lam and Qingquan Li made important comments and suggestions for improvement of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Chen, B.Y.; Lam, W.H.K.; Sumalee, A.; Li, Q.Q.; Shao, H.; Fang, Z.X. Finding reliable shortest paths in road networks under uncertainty. *Netw. Spat. Econ.* **2013**, *13*, 123–148. [CrossRef]
- Chen, B.Y.; Li, Q.Q.; Lam, W.H.K. Finding the k reliable shortest paths under travel time uncertainty. *Transp. Res. B Methodol.* **2016**, *94*, 189–203. [CrossRef]
- Yang, L.; Zhou, X. Optimizing on-time arrival probability and percentile travel time for elementary path finding in time-dependent transportation networks: Linear mixed integer programming reformulations. *Transp. Res. B Methodol.* **2017**, *96*, 68–91. [CrossRef]
- Zhong, R.; Sumalee, A.; Maruyama, T. Dynamic marginal cost, access control, and pollution charge: A comparison of bottleneck and whole link models. *J. Adv. Transp.* **2012**, *46*, 191–221. [CrossRef]
- Zhao, J.; Ma, W.; Liu, Y.; Han, K. Optimal operation of freeway weaving segment with combination of lane assignment and on-ramp signal control. *Transp. A* **2016**, *12*, 413–435. [CrossRef]
- Chen, B.Y.; Yuan, H.; Li, Q.Q.; Shaw, S.L.; Lam, W.H.K.; Chen, X. Spatiotemporal data model for network time geographic analysis in the era of big data. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 1041–1071. [CrossRef]
- Lim, S.; Lee, C. Data fusion algorithm improves travel time predictions. *IET Intell. Transp. Syst.* **2011**, *5*, 302–309. [CrossRef]
- Mori, U.; Mendiburu, A.; Alvarez, M.; Lozano, J.A. A review of travel time estimation and forecasting for Advanced Traveller Information Systems. *Transp. A* **2015**, *11*, 119–157. [CrossRef]
- Chen, B.Y.; Yuan, H.; Li, Q.Q.; Lam, W.H.K.; Shaw, S.L.; Yan, K. Map matching algorithm for large-scale low-frequency floating car data. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 22–38. [CrossRef]
- Du, L.; Peeta, S.; Kim, Y.H. An adaptive information fusion model to predict the short-term link travel time distribution in dynamic traffic networks. *Transp. Res. B Methodol.* **2012**, *46*, 235–252. [CrossRef]
- Bachmann, C.; Abdulhai, B.; Roorda, M.J.; Moshiri, B. A comparative assessment of multi-sensor data fusion techniques for freeway traffic speed estimation using microsimulation modeling. *Transp. Res. C Emerg. Technol.* **2013**, *26*, 33–48. [CrossRef]
- Bachmann, C.; Roorda, M.J.; Abdulhai, B.; Moshiri, B. Fusing a bluetooth traffic monitoring system with loop detector data for improved freeway traffic speed estimation. *J. Intell. Transp. Syst.* **2013**, *17*, 152–164. [CrossRef]
- Deng, W.; Lei, H.; Zhou, X. Traffic state estimation and uncertainty quantification based on heterogeneous data sources: A three detector approach. *Transp. Res. B Methodol.* **2013**, *57*, 132–157. [CrossRef]
- Tam, M.L.; Lam, W.H.K. Using automatic vehicle identification data for travel time estimation in Hong Kong. *Transportmetrica* **2008**, *4*, 179–194. [CrossRef]
- Zou, H.; Yue, Y.; Li, Q.; Yeh, A.G.O. An improved distance metric for the interpolation of link-based traffic data using kriging: A case study of a large-scale urban road network. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 667–689. [CrossRef]

16. El Esawey, M.; Sayed, T. Travel time estimation in urban networks using limited probes data. *Can. J. Civil. Eng.* **2011**, *38*, 305–318. [CrossRef]
17. Liu, H.X.; Ma, W. A virtual vehicle probe model for time-dependent travel time estimation on signalized arterials. *Transp. Res. C Emerg. Technol.* **2009**, *17*, 11–26. [CrossRef]
18. Liu, H.X.; Ma, W.; Wu, X.; Hu, H. Real-time estimation of arterial travel time under congested conditions. *Transportmetrica* **2012**, *8*, 87–104. [CrossRef]
19. Ndoye, M.; Totten, V.F.; Krogmeier, J.V.; Bullock, D.M. Sensing and signal processing for vehicle reidentification and travel time estimation. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 119–131. [CrossRef]
20. Yu, B.; Lam, W.H.K.; Tam, M.L. Bus arrival time prediction at bus stop with multiple routes. *Transp. Res. C Emerg. Technol.* **2011**, *19*, 1157–1170. [CrossRef]
21. Vlahogianni, E.I.; Golias, J.C.; Karlaftis, M.G. Short-term traffic forecasting: Overview of objectives and methods. *Transp. Rev.* **2004**, *24*, 533–557. [CrossRef]
22. El Faouzi, N.E. Data fusion in road traffic engineering: An overview. In Proceedings of the SPIE—The International Society for Optical Engineering, Orlando, FL, USA, 14–15 April 2004.
23. Choi, K.; Chung, Y. A data fusion algorithm for estimating link travel time. *J. Intell. Transp. Syst.* **2002**, *7*, 235–260. [CrossRef]
24. El Faouzi, N.E. Bayesian and evidential approaches for traffic data fusion: Methodological issues and case study. In Proceedings of the Transportation Research Board 85th Annual Meeting (No. 06–1510), Washington, DC, USA, 22–26 January 2006.
25. El Faouzi, N.E.; Klein, L.A.; De Mouzon, O. Improving travel time estimates from inductive loop and toll collection data with Dempster-Shafer data fusion. *Transport. Res. Rec.* **2009**, *2129*, 73–80. [CrossRef]
26. Kong, Q.J.; Li, Z.; Chen, Y.; Liu, Y. An approach to urban traffic state estimation by fusing multisource information. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 499–511. [CrossRef]
27. Kong, Q.J.; Chen, Y.; Liu, Y. A fusion-based system for road-network traffic state surveillance: A case study of Shanghai. *IEEE Intell. Transp. Syst.* **2009**, *1*, 37–42. [CrossRef]
28. Nantes, A.; Dong, N.; Bhaskar, A.; Miska, M.; Chung, E. Real-time traffic state estimation in urban corridors from heterogeneous data. *Transp. Res. C Emerg. Technol.* **2015**, *66*, 99–118. [CrossRef]
29. Shan, Z.; Xia, Y.; Hou, P.; He, J. Fusing Incomplete Multisensor Heterogeneous Data to Estimate Urban Traffic. *IEEE Multimed.* **2016**, *23*, 56–63. [CrossRef]
30. Lederman, R.; Wynter, L. Real-time traffic estimation using data expansion. *Transp. Res. B Methodol.* **2011**, *45*, 1062–1079. [CrossRef]
31. Haworth, J.; Cheng, T. Non-parametric regression for space-time forecasting under missing data. *Comput. Environ. Urban.* **2012**, *36*, 538–550. [CrossRef]
32. Li, L.; Li, Y.B.; Li, Z.H. Efficient missing data imputing for traffic flow by considering temporal and spatial dependence. *Transp. Res. C Emerg. Technol.* **2013**, *34*, 108–120. [CrossRef]
33. Chan, K.S.; Lam, W.H.K.; Tam, M.L. Real-time estimation of arterial travel times with spatial travel time covariance relationships. *Transp. Res. Rec.* **2009**, *2121*, 102–109. [CrossRef]
34. Dion, F.; Rakha, H. Estimating dynamic roadway travel times using automatic vehicle identification data for low sampling rates. *Transp. Res. B Methodol.* **2006**, *40*, 745–766. [CrossRef]
35. Jenelius, E.; Koutsopoulos, H.N. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transp. Res. B Methodol.* **2013**, *53*, 64–81. [CrossRef]
36. Rahmani, M.; Jenelius, E.; Koutsopoulos, H.N. Non-parametric estimation of route travel time distributions from low-frequency floating car data. *Transp. Res. C Emerg. Technol.* **2015**, *58*, 343–362. [CrossRef]
37. Hans, E.; Chiabaut, N.; Leclercq, L. Applying variational theory to travel time estimation on urban arterials. *Transp. Res. B Methodol.* **2015**, *78*, 169–181. [CrossRef]
38. Dempster, A.P. Upper and lower probabilities induced by multi-valued mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339. [CrossRef]
39. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
40. Beynon, M.; Cosker, D.; Marshall, D. An expert system for multi-criteria decision making using Dempster Shafer theory. *Expert Syst. Appl.* **2001**, *20*, 357–367. [CrossRef]
41. Hegarat-Masclé, S.L.; Richard, D.; Otle, C. Multi-scale data fusion using Dempster-Shafer evidence theory. *Integr. Comput. Aided Eng.* **2003**, *10*, 9–22.

42. Gong, Y.; Wang, Y. Application Research on Bayesian Network and D-S Evidence Theory in Motor Fault Diagnosis. In Proceedings of the 6th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), Shenyang, China, 1–3 November 2013.
43. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *14*, 28–44. [CrossRef]
44. Deng, Y.; Chan, F.T.S. A new fuzzy dempster MCDM method and its application in supplier selection. *Expert Syst. Appl.* **2011**, *38*, 9854–9861. [CrossRef]
45. Su, S.Y.; Deng, Y.; Mahadevan, S.; Bao, Q.L. An improved method for risk evaluation in failure modes and effects analysis of aircraft engine rotor blades. *Eng. Fail. Anal.* **2012**, *26*, 164–174. [CrossRef]
46. Parikh, C.R.; Pont, M.J.; Jones, N.B. Application of Dempster–Shafer theory in condition monitoring applications: A case study. *Pattern Recogn. Lett.* **2001**, *22*, 777–785. [CrossRef]
47. Dou, Z.; Xu, X.; Lin, Y.; Zhou, R. Application of D-S evidence fusion method in the fault detection of temperature sensor. *Math. Probl. Eng.* **2014**, *2014*, 1–6. [CrossRef]
48. Fan, X.; Zuo, M.J. Fault diagnosis of machines based on D-S evidence theory. Part 1: D–S evidence theory and its improvement. *Pattern Recogn. Lett.* **2006**, *27*, 366–376. [CrossRef]
49. Hu, Y.; Fan, X.; Zhao, H.; Hu, B. The Research of Target Identification Based on Neural Network and D-S Evidence Theory. In Proceedings of the International Asia Conference on Informatics in Control, Bangkok, Thailand, 1–2 February 2009.
50. Dong, G.; Kuang, G. Target recognition via information aggregation through Dempster–Shafer’s evidence theory. *IEEE Geosci. Remote Sens.* **2015**, *12*, 1247–1251. [CrossRef]
51. Li, Y.; Chen, J.; Ye, F.; Liu, D. The improvement of DS evidence theory and its application in IR/MMW target recognition. *J. Sens.* **2016**, *2016*, 1–15. [CrossRef]
52. Dymova, L.; Sevastjanov, P. An Interpretation of Intuitionistic Fuzzy Sets in the Framework of the Dempster-Shafer Theory: Decision making aspect. *Knowl. Based Syst.* **2010**, *23*, 772–782. [CrossRef]
53. Chen, N.; Sun, F.; Ding, L.; Wang, H. An adaptive PNN-DS approach to classification using multi-sensor information fusion. *Neural Comput. Appl.* **2009**, *18*, 455–467. [CrossRef]
54. Yager, R.R. On the Dempster-Shafer framework and new combination rules. *Inf. Sci.* **1987**, *41*, 93–137. [CrossRef]
55. Smets, P. The Combination of Evidence in the Transferable Belief Model. *IEEE Trans. Pattern Anal.* **1990**, *12*, 447–458. [CrossRef]
56. Murphy, C.K. Combining belief functions when evidence conflicts. *Decis. Support Syst.* **2000**, *29*, 1–9. [CrossRef]
57. Chen, B.Y.; Shi, C.; Zhang, J.; Lam, W.H.K.; Li, Q.Q.; Xiang, S. Most reliable path-finding algorithm for maximizing on-time arrival probability. *Transp. B* **2016**, *5*, 204–221. [CrossRef]
58. Lomax, T.; Schrank, D.; Turner, S.; Margiotta, R. *Selecting Travel Reliability Measures*; Texas Transportation Institute Monograph: College Station, TX, USA, 2003.
59. Hart, R.G. A Close approximation related to the error function. *Math. Comput.* **1966**, *20*, 600–602. [CrossRef]
60. Khosravi, A.; Mazloui, E.; Nahavandi, S.; Creighton, D.; Van Lint, J.W.C. A genetic algorithm-based method for improving quality of travel time prediction intervals. *Transp. Res. C Emerg. Technol.* **2011**, *19*, 1364–1376. [CrossRef]
61. Khosravi, A.; Mazloui, E.; Nahavandi, S.; Creighton, D.; Van Lint, J.W.C. Prediction intervals to account for uncertainties in travel time prediction. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 537–547. [CrossRef]
62. Shi, C.; Chen, B.Y.; Li, Q. Estimation of Travel Time Distributions in Urban Road Networks Using Low-Frequency Floating Car Data. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 253. [CrossRef]



Article

A General Framework for 3-D Parameters Estimation of Roads Using GPS, OSM and DEM Data

Christophe Boucher ^{*,†,‡} and Jean-Charles Noyer [‡]

Laboratoire d'Informatique Signal et Image de la Côte d'Opale, Université du Littoral Côte d'Opale, 59183 Dunkerque, France; jean-charles.noyer@univ-littoral.fr

* Correspondence: christophe.boucher@univ-littoral.fr; Tel.: +33-3214-65658

† Current address: 50 rue Ferdinand Buisson, BP 719, 62228 Calais CEDEX, France.

‡ These authors contributed equally to this work.

Received: 11 November 2017; Accepted: 19 December 2017; Published: 25 December 2017

Abstract: A growing number of applications needs GIS mapping information and commercial 3-D roadmaps especially. This paper presents a solution of accessing freely to 3-D map information and updating in the context of transport applications. The method relies on the OSM road networks that is 2-D modeled intrinsically. The objective is to estimate the road elevation and inclination parameters by fusing GPS, OSM and DEM data through a nonlinear filter. An experimental framework, using ASTER GDEM2 data, shows some results of the improvement of the roads modeling that includes their slopes also. The map database can be enriched with the estimated inclinations. The accuracy depends on the GPS and DEM elevation errors (typically a few meters with the GNSS sensors used and the DEM under consideration).

Keywords: multi-sensor fusion; non-linear filtering; GNSS-based navigation; land-vehicle localization; digital road maps; digital elevation models

1. Introduction

Nowadays, an increasing number of transport applications would need to have access to 3-D map information whether for location or route planning problems [1]. Some solutions of 3-D maps exist but they are commercial and expensive. In this context, the purpose of our work is to offer a free solution based on the use of OpenStreetMap (OSM, <http://www.openstreetmap.org>). OSM is a free web-service that provides map data from uploaded GPS traces and user-registered information as names, types, traffic directions and inclinations of roads [2]. More generally, OSM is built by volunteer users who contribute and maintain these GIS data all over the world. The level of accuracy depends primarily on existing data. It should be noticed that the level of accuracy is optimal when one is in a dense area for that many users have refined the data over time. Providing none 3-D support, the OSM database is increasingly used in the development of transport applications [3,4]. In our earlier work [5], a method has been proposed to update the OSM roads slopes by using a car positioning system composed of 2 GPS units. To make the method more robust, we are interested in the fusion of the dual-GPS positioning system and data from a Digital Elevation Model (DEM) that accounts for the terrain relief. This can lead to the improvement of the estimated elevations [6]. One of the objectives of the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) mission was to produce a global DEM of a very large part of the Earth, based on a horizontal resolution of 1 arcsecond (30 m) [7]. In 2009, this American-Japanese collaboration allows to produce a model whose horizontal resolution was three times sharper than the DEM of the Shuttle Radar Topography Mission (SRTM) [8] that was built with at 3 arcseconds (about 90 m). However, the presence of many voids and artifacts in this first model version made the exploitation of data quite difficult. In 2011, a new model version was proposed with a significant refinement of the relief. Some artifacts as steps at scene

boundaries, pits, bumps or mole runs are fixed without deleting all of them. Therefore, 1×1 degree tiles of data are available with additional information of the elevation measurements quality. The WGS84 horizontal datum is used to locate ASTER data while the EGM96 vertical datum levels the stored elevations compared to the Mean Sea Level (MSL) geoid. Several authors published works about the relevance and the accuracy of DEM data in comparison with calibrated GPS-altitudes [9,10]. Considering the horizontal resolution of ASTER DEM data, an elevation information is available about every 30 m. This low spatial sampling can be a major drawback of the ASTER data. In a process of estimating the slope of roads, it is hard to achieve a high level of accuracy using ASTER DEM only [11]. The possibility to use DEM only solutions is indeed related to the spatial dimensions of the tiles that are for now not small enough. Three-dimensional modeling of road networks has resulted in many publications of methods based on the processing of aerial images [12], geometric methods using 2D GIS data only [13], or methods based on the fusion of GPS/GIS/DEM data [14–17].

This paper proposes to enrich the modeling of road network with an additional parameter related to the inclination and the vertical height of roads. By fusing GPS/DEM data through a nonlinear filter (typically an unscented Kalman filter [18]), our method allows to get an estimation of the OSM roads slopes. The aim is to complete the mapping system with 3-D information as nodes elevations and inclinations of roads. We equipped a standard vehicle with a Global Navigation Satellite System (GNSS) based on a dual-GPS system whose receivers antennas are aligned on the extremities of the same side of the roof. A such consideration induces a more accurate positioning of the vehicle by bounding the possible errors due to GPS signal degradation [19]. Knowing that OSM provides a digital modeling of roads that relies of the 2-D geolocation of their nodes depending on land surveys, the method requires a map-matching step with GPS data. The goal is to deliver an estimation of the road inclinations. To this end, we develop an original method based on a geometric approach to manage the DEM data and compute a more accurate MSL-elevation. In the proposed solution, all measurements are sequentially processed by an unscented Kalman filter (UKF) that is more suited to nonlinear equations than the standard Kalman filter. An original aspect of our approach lies in the modeling of the OSM and DEM databases. They are considered as sensors with their own measurement errors which makes possible to develop a global filtering approach. Finally, by injecting the estimated vehicle elevations and their orthogonal projections on the matched road into a least squares method, we get an estimate of the slope of this road.

We have chosen to validate this approach in a context of road experimentation mainly composed of highways that are built in an open environment with a varied but limited terrain elevation. We present experimental results of 3-D time-location of the vehicle that allow to get an accurate estimation of the inclination of the roads traveled during the test. Then, it becomes possible to register them in OSM for later use in 3-D route planning applications, as developed in [20].

2. 3-D Modeling

Stating the standard kinematic equations are enough to model the car dynamics, the major parameters are defined by the acceleration γ , the velocity v and the position. They are reported in the following state vector by introducing their x , y and z -coordinates along the 3-D axes:

$$X_t = (x_t, v_t^x, \gamma_t^x, y_t, v_t^y, \gamma_t^y, z_t, v_t^z, \gamma_t^z)^T \quad (1)$$

2.1. Formatting of the Dynamics Equation

Introducing the u -axis with $u = \{x, y, z\}$, the car dynamics relies on the following third order kinematic state model:

$$\begin{pmatrix} u_{t+1} \\ v_{t+1}^u \\ \gamma_{t+1}^u \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix}}_{F_u} \begin{pmatrix} u_t \\ v_t^u \\ \gamma_t^u \end{pmatrix} + \begin{pmatrix} \Omega_t^u \\ \Omega_t^v \\ \Omega_t^\gamma \end{pmatrix} \tag{2}$$

The dynamics matrix F_u reveals the navigation rate Δt . The Ω -vector is composed of additive white Gaussian noises with $\Omega_t \sim \mathcal{N}(0, Q_t)$. This equation can be generalized to the three-dimensional case and becomes:

$$X_{t+1} = f(X_t, \Omega_t) = FX_t + \Omega_t \tag{3}$$

F models the linear flow of the dynamics. Stating $\mathbb{O}_{3 \times 3}$, the null square-matrix of dimension 3, it can be expanded as:

$$F = \begin{pmatrix} F_x & \mathbb{O}_{3 \times 3} & \mathbb{O}_{3 \times 3} \\ \mathbb{O}_{3 \times 3} & F_y & \mathbb{O}_{3 \times 3} \\ \mathbb{O}_{3 \times 3} & \mathbb{O}_{3 \times 3} & F_z \end{pmatrix} \tag{4}$$

2.2. Formatting the Measurement Equations

2.2.1. GPS Readings

The car carries two EGNOS-aided (SBAS) receivers that collect the GPS positioning (the specifications of the GPS sensors are given in Section 5.1). Their antennas are fixed on the roof and are spaced by a distance D . $Z_{t+1}^{GPS,i}$ denotes the measurement that is providing by the receiver $i = \{1, 2\}$. Both them are fused sequentially to get an estimation of the 3-D dynamics:

$$Z_{t+1}^{GPS,i} = \begin{pmatrix} x_{t+1}^{GPS,i} \\ y_{t+1}^{GPS,i} \\ z_{t+1}^{GPS,i} \end{pmatrix} = h^{GPS,i}(X_{t+1}) + \mathcal{V}_{t+1}^{GPS,i} \tag{5}$$

$\mathcal{V}_{t+1}^{GPS,i} \sim \mathcal{N}(0, R_{t+1}^{GPS})$ is an additive white Gaussian noise with zero mean and R_{t+1}^{GPS} as the noise covariance matrix.

A single coordinate system is used to define the car positioning. It refers to the GPS antenna of unit #1. Taking into account this consideration, the measurement function of each GPS unit can be formatted as follows:

- for the GPS unit #1, $h^{GPS,1}$ is the linear measurement function:

$$h^{GPS,1}(X_{t+1}) = \begin{pmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \end{pmatrix} \tag{6}$$

- for the GPS unit #2, $h^{GPS,2}$ is the nonlinear measurement function:

$$h^{GPS,2}(X_{t+1}) = \begin{pmatrix} x_{t+1} + D * \frac{v_{t+1}^x}{\|v_{t+1}\|} \\ y_{t+1} + D * \frac{v_{t+1}^y}{\|v_{t+1}\|} \\ z_{t+1} + D * \frac{v_{t+1}^z}{\|v_{t+1}\|} \end{pmatrix} \tag{7}$$

Here, the introduced non-linearities are derived from the calculations making it possible to define the GPS measurement $\mathcal{Z}_{t+1}^{\text{GPS},2}$ in the coordinate system that is fitted to the antenna of the receiver #1.

2.2.2. OSM Road Network

The next step is to fuse the previous GPS-estimated parameters with the OSM database in order to match the traveled road. As a consequence, the accuracy of the car ground-positioning can be increased since the OSM road network is modeled by road segments whose 2-D location of transitional nodes from one segment to another is available. The accuracy of the GPS-based positioning is straightforwardly related to the intrinsic performance of the GPS receiver (e.g., about a few meters for our sensors). Fusing GPS-based estimation with the OSM data allows to consider the road geometry as a constraint for the car displacement. In this case, the estimator can take benefit of the accuracy of the map database. The ground-location $(x_{t+1}^{\text{OSM}}, y_{t+1}^{\text{OSM}})$ and the ground-direction of its road segment $\theta_{t+1}^{\text{OSM}}$ are the main features of each node:

$$\mathcal{Z}_{t+1}^{\text{OSM}} = \begin{pmatrix} x_{t+1}^{\text{OSM}} \\ y_{t+1}^{\text{OSM}} \\ \theta_{t+1}^{\text{OSM}} \end{pmatrix} = h^{\text{OSM}}(X_{t+1}) + \mathcal{V}_{t+1}^{\text{OSM}} \quad (8)$$

$\mathcal{V}_{t+1}^{\text{OSM}}$ is an additive white Gaussian noise with zero mean and R_{t+1}^{OSM} as the noise covariance matrix so that the statistical aspects related to the inhomogeneities and errors of land surveys can be integrated. In order to link the stater vector components to the measurements, a new nonlinear function h^{OSM} is introduced as:

$$h^{\text{OSM}}(X_{t+1}) = \begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \arctan\left(\frac{\sigma_{t+1}^y}{\sigma_{t+1}^x}\right) \end{pmatrix} \quad (9)$$

When the vehicle is stopped, the previous direction is reused.

2.2.3. ASTER GDEM2 Terrain Surface

The final step is the identification of the DEM-area in which the car is driven. Once matched the MSL-elevation is fused with the previous OSM-estimated parameters by the proposed algorithm that takes into account the ASTER GDEM2 horizontal resolution of 1 arcsecond (about 30m). The DEM measure features the ground-location of the MSL-elevation point as $(x_{t+1}^{\text{DEM}}, y_{t+1}^{\text{DEM}}, z_{t+1}^{\text{DEM}})$.

$$\mathcal{Z}_{t+1}^{\text{DEM}} = \begin{pmatrix} x_{t+1}^{\text{DEM}} \\ y_{t+1}^{\text{DEM}} \\ z_{t+1}^{\text{DEM}} \end{pmatrix} = h^{\text{DEM}}(X_{t+1}) + \mathcal{V}_{t+1}^{\text{DEM}} \quad (10)$$

h^{DEM} is the measurement function and $\mathcal{V}_{t+1}^{\text{DEM}}$ is an additive white Gaussian noise with zero mean and R_{t+1}^{DEM} as the noise covariance matrix. This matrix defines the standard deviations relative to the horizontal position and vertical elevation measurement.

$$h^{\text{DEM}}(X_{t+1}) = \begin{pmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} - H \end{pmatrix} \quad (11)$$

This function highlights the term H that defines the height that has been measured initially between GPS antenna of the unit #1 and the ground.

3. Estimation of the Road Inclination

The goal of this paper is to propose a theoretical framework to OSM data enrichment with road inclination. The proposed nonlinear modeling is solved by an unscented Kalman filter (UKF). It is known that the UKF approach improves the state estimation compared with the EKF-methods in the nonlinear case [21]. The benefit of the UKF-approach also lies in its lower computational complexity compared with the particle filtering methods [22] which are not essential here due to moderate non-linearity of the modeling. Figure 1 shows that the filter is based on a sequential processing structure of the GPS, OSM and DEM data. Finally, the filter estimates the inclination which is used to enrich the road network database.

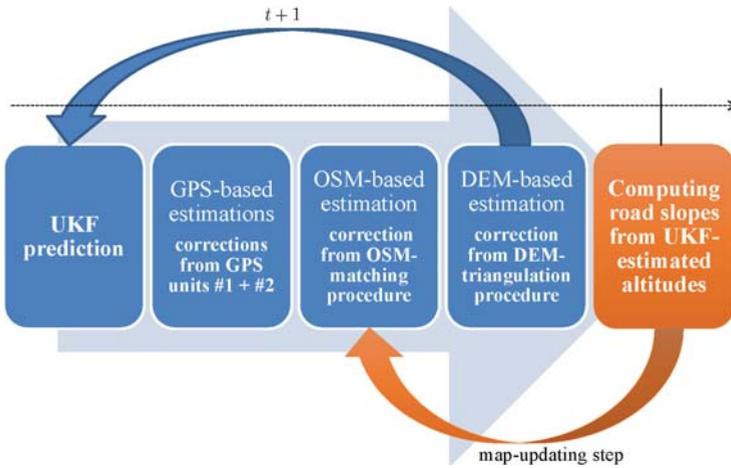


Figure 1. Multisensor fusion structure.

3.1. Dynamics Sampling: Prediction

A first step consists in a statistical sampling of the state-space vector using sigma-points. Let $\chi_{t|t}^{j,x}$ and $\chi_{t|t}^{j,\Omega}$ be the sigma-points at time step t based on the knowledge of the measurement at time step t , respectively related to the state vector X and the dynamics noise Ω .

Their evolution is computed as follows:

$$\chi_{t+1|t}^{j,x} = f(\chi_{t|t}^{j,x}, \chi_{t|t}^{j,\Omega}) \tag{12}$$

$\chi_{t+1|t}^{j,x}$ is the j^{th} column of $\chi_{t+1|t}^x$. The predicted state is computed as follows:

$$\hat{X}_{t+1|t} = \sum_{j=1}^{2n_a+1} w_j \chi_{t+1|t}^{j,x} \tag{13}$$

n_a is the augmented state vector dimension and the weights w_j are derived from the unscented transform [23]. The covariance matrix $\tilde{P}_{t+1|t}$ is written as below:

$$\tilde{P}_{t+1|t} = \sum_{j=1}^{2n_a+1} w_j (\chi_{t+1|t}^{j,x} - \hat{X}_{t+1|t})(\chi_{t+1|t}^{j,x} - \hat{X}_{t+1|t})^T \tag{14}$$

3.2. Correction: General Principles

Let us denote Z^α the measurement related to the sensor α where $\alpha \in \{\text{GPS}_1, \text{GPS}_2, \text{OSM}, \text{DEM}\}$. The measurement equation can be written as:

$$Z_{t+1}^\alpha = h^\alpha(X_t) + \mathcal{V}_{t+1}^\alpha \tag{15}$$

From a general viewpoint, the correction step use the sigma-points $\chi_{t+1|t}^{j,z^\alpha}$ related to the measurement of the sensor α :

$$\chi_{t+1|t}^{z^\alpha} = h^\alpha(\chi_{t+1|t}^x) + \chi_{t+1|t}^{y^\alpha} \tag{16}$$

The predicted measurement $\hat{Z}_{t+1|t}^\alpha$ is computed as the weighted sum of the sigma-points $\chi_{t+1|t}^{j,z^\alpha}$ where the superscript j denotes the j^{th} column of $\chi_{t+1|t}^{z^\alpha}$:

$$\hat{Z}_{t+1|t}^\alpha = \sum_{j=1}^{2n_a+1} w_j \chi_{t+1|t}^{j,z^\alpha} \tag{17}$$

The state vector and the error covariance matrix are computed as follows:

$$\begin{cases} K_{t+1}^\alpha &= \tilde{P}_{X_{t+1}Z_{t+1}^\alpha} \tilde{P}_{Z_{t+1}^\alpha Z_{t+1}^\alpha}^{-1} \\ \tilde{P}_{t+1|t+1}^\alpha &= \tilde{P}_{t+1|t}^\alpha - K_{t+1}^\alpha \tilde{P}_{Z_{t+1}^\alpha Z_{t+1}^\alpha} K_{t+1}^{\alpha T} \\ \hat{X}_{t+1|t+1}^\alpha &= \hat{X}_{t+1|t}^\alpha + K_{t+1}^\alpha (Z_{t+1}^\alpha - \hat{Z}_{t+1|t}^\alpha) \end{cases} \tag{18}$$

where

$$\tilde{P}_{X_{t+1}Z_{t+1}^\alpha} = \sum_{j=1}^{2n_a+1} w_j (\chi_{t+1|t}^{j,x} - \hat{X}_{t+1|t}^\alpha)(\chi_{t+1|t}^{j,z^\alpha} - \hat{Z}_{t+1|t}^\alpha)^T \tag{19}$$

$$\tilde{P}_{Z_{t+1}^\alpha Z_{t+1}^\alpha} = \sum_{j=1}^{2n_a+1} w_j (\chi_{t+1|t}^{j,z^\alpha} - \hat{Z}_{t+1|t}^\alpha)(\chi_{t+1|t}^{j,z^\alpha} - \hat{Z}_{t+1|t}^\alpha)^T \tag{20}$$

3.3. Correction: Proposed Solution

The proposed solution is based on a sequential processing of sensor data. An original feature of our method consists in a statistical modeling of map data from OSM and DEM. The popular approaches match the estimate of the position with the road network [24]. Our statistical method allows to take into account the mapping inaccuracies in the estimation process. Indeed, in many solutions, the map matching procedure uses a projection onto the road network and thus does not consider the map accuracy as a parameter of the problem. Our approach uses this accuracy to build a map-matching procedure that gives in the estimation process a relative weight to the OSM-based correction. This is done via the measurement covariance matrix R_t^{OSM} . The accuracy of the OSM measurements is relative to the accuracy of the GPS traces uploaded by the users in the database.

Moreover, the DEM database, also modeled as an additional sensor, is used to correct the estimation of the elevation of the vehicle. Similarly to OSM measurements, the DEM data are used in the estimation process, considering their accuracy defined by the covariance matrix R_t^{DEM} . Again with this "sensor", we do not use a all-or-nothing correction but a weighted correction using the DEM measurement covariance matrix. This leads to an improvement of the estimation accuracy, especially of the elevation which remains a parameter with high errors in GPS (typically several meters in some cases or more in multipath propagation problems of GNSS signals).

As a consequence, the integration of OSM and DEM data leads to a better estimation of the road network parameters and in particular of the road sections elevation (including slope). This estimation

is performed in real time and improves the OSM database by extending it to the 3-D case. The updated database will then be used for future positioning thus leading to an overall improvement in estimation.

Figure 1 details the different steps of the filtering solution. The measurements are sequentially processed in the following order: GPS readings #1 and #2, OSM measurements and DEM measurements. The relative weight of each sensor is related to their measurement error covariance matrix which initially appears in the computation of the sigma-points $\chi_{t+1|t}^{\mathcal{Z}^s}$ related to measurement equation Equation (16).

3.3.1. Correction from GPS Sensors

The GPS readings of each receiver # i ($i = \{1, 2\}$) are sequentially processed to estimate the state vector $\hat{X}_{t+1/t+1}^{\text{GPS}_i}$.

3.3.2. Correction from the OSM Database

The OSM database is modeled as an additional sensor (see Equation (16)). The prediction of the map measurement $\hat{\mathcal{Z}}_{t+1|t+1}^{\text{OSM}}$ is done from the unscented transform (Equation (17)). This prediction is matched with the OSM road network and the matched segment is used to correct the estimation of the ground-location and direction of the vehicle.

In [24], the author lists various approaches for map-aided positioning of a vehicle and most of them concerns geometric methods. We propose to solve the map matching step through a statistical approach that relies on the computing of the Mahalanobis distance [25] between the UKF-predicted map measurement and all available OSM road descriptors. Knowing that the digital road map is described by geolocated and interconnected nodes whose main drawback is their random sampling grid, we have refined the map modeling. In the matching step, we introduce as inputs the orthogonal projections of the UKF-predicted map measurement onto each nearby OSM road segment if possible, otherwise the nearest line segment extremity is used instead of the projection. This step leads to the identification of the position of the closest candidate for each road segment of the map database.

At time instant $t + 1$, we get the set of orthogonal projections $\mathcal{Z}_m^{\text{OSM}}$:

$$\mathcal{Z}_m^{\text{OSM}} = \text{proj}_{\perp, m}(\hat{\mathcal{Z}}_{t+1|t+1}^{\text{OSM}}) \quad (21)$$

$m = \{1, \dots, n\}$ and n describes the maximum projections number. Map descriptor includes the road direction. So, the Mahalanobis distance $d_m = d(\mathcal{Z}_m^{\text{OSM}}, \hat{\mathcal{Z}}_{t+1|t+1}^{\text{OSM}})$ can be evaluated in the cases of a two-way road (θ_1, θ_2) or a one-way road (θ_1) :

$$\begin{cases} \mathcal{Z}_m^{\text{OSM}, \theta_1} = (x_m^{\text{OSM}}, y_m^{\text{OSM}}, \theta_m^{\text{OSM}})^T \\ \mathcal{Z}_m^{\text{OSM}, \theta_2} = (x_m^{\text{OSM}}, y_m^{\text{OSM}}, \theta_m^{\text{OSM}} + \pi)^T \end{cases} \quad (22)$$

The Mahalanobis distances can be derived from the previous equation:

$$(d_m^{\theta_k})^2 = (\mathcal{Z}_m^{\text{OSM}, \theta_k} - \hat{\mathcal{Z}}_{t+1|t+1}^{\text{OSM}})^T (\tilde{P}_{t+1|t+1}^{\text{GPS}})^{-1} (\mathcal{Z}_m^{\text{OSM}, \theta_k} - \hat{\mathcal{Z}}_{t+1|t+1}^{\text{OSM}}) \text{ with } k = \{1, 2\} \quad (23)$$

The innovation process has a Gaussian distribution. As a consequence, the squared Mahalanobis distance is then distributed according to a chi-square law with three degrees of freedom. The correction is done if the value is below a threshold ε that results from the statistical properties of the Mahalanobis distance. Indeed, the measurement $\mathcal{Z}_{m^*}^{\text{OSM}, \theta^*}$ that minimizes the criterion $d_m^{\theta}(\mathcal{Z}_m^{\text{OSM}}, \hat{\mathcal{Z}}_{t+1|t+1}^{\text{OSM}}) \leq \varepsilon$ is used to correct the GPS-based estimate.

3.3.3. Correction from DEM Database

A DEM database is used to take into account the elevation in the multisensor processing. A first solution has been proposed in [26] and was based on the minimization of Mahalanobis distances

between the nearest horizontal point of elevation and the filter estimation. In area of low elevation change, this approach may be sufficient. In areas with higher elevation, the use of raw DEM data can generate significant errors. In Section 4, we develop our main contribution that is relative to a new modeling of DEM data based on triangulations.

The last step concerns the correction with elevation measurements from the ASTER terrain surface. The goal is to bound with DEM corrections [27], the well-known altitude errors of GPS measurements [28]. The position and elevation errors of the DEM data are modeled by Equation (10) and the unscented transform is used to derive the predicted measurement. It must be noticed that the elevation components only are corrected to avoid horizontal positioning errors that can be caused by the poor horizontal resolution of the DEM. So, the measurement Z_0^{DEM} is used to correct the previous estimate and deliver the final estimate:

$$\hat{X}_{t+1|t+1} = \hat{X}_{t+1|t+1}^{DEM} \tag{24}$$

3.4. Road Slope Computing

The final objective of this work is to propose an enrichment of OSM database with the inclinations of the roads. This step is done from the estimated elevation (z-component of the state vector in Equation (24)) and the matched orthogonal projection on the OSM road network (see Equation (21)). The last step relies then on the estimation of the inclination parameters.

The following equation details the relationship between the elevation $E_{s,t}$ and the radial distance $\rho_{s,t}$ of the projected estimation at t onto the line segment # s to the extremity $(x_{s,0}, y_{s,0})$:

$$E_{s,t} = a_s \rho_{s,t} + z_{s,0} \tag{25}$$

where a_s denotes the slope of the line segment # s and $z_{s,0}$ its intercept.

For the n_s projections onto the same line segment # s , Equation (25) can be written as follows:

$$E_s = \mathcal{H}u = \begin{pmatrix} \rho_{s,1} & 1 \\ \vdots & \vdots \\ \rho_{s,n_s} & 1 \end{pmatrix} \begin{pmatrix} a_s \\ z_{s,0} \end{pmatrix} \tag{26}$$

where the radial distance $\rho_{s,t}$ is described by:

$$\rho_{s,t} = \sqrt{(x_{s,t}^{OSM} - x_{s,0})^2 + (y_{s,t}^{OSM} - y_{s,0})^2}, \quad \forall t \tag{27}$$

\mathcal{H} is a $n_s \times 2$ matrix. The parameter n_s changes for each line segment # s and depends on several parameters (vehicle speed, length of the line segment, ...).

For each segment, a least squares method is used to estimate the road inclination:

$$\begin{pmatrix} \hat{a}_s \\ \hat{z}_{s,0} \end{pmatrix} = (\mathcal{H}^T \mathcal{H})^{-1} \mathcal{H}^T E_s \tag{28}$$

This statistical estimation of the inclination is used to automatically enrich the road map database.

4. New Modeling of DEM Data

We propose a new modeling of DEM-data based on the triangulations in the vicinity of the ground-location of the vehicle, instead of matching raw data as published in [26]. So, the problem is now to estimate the elevation between two successive grid points. This method allows to compute the pseudo-elevation of the vehicle from the DEM grid. Here, the new advantage is that the DEM management method takes into account the topology of all terrain.

4.1. Triangulation of the DEM

Another solution can be to decompose the grid into triangles. We use the rasterized DEM to produce a triangulated irregular network (TIN). This modeling produce a continuous surface that can be used in the filtering solution in a similar way to OSM road network (which is a line segment network). This modeling avoids the problem of estimating planes equations using a least squares method which can lead to difficulties in high variability of the orientation of the plane (for an approach to 4 points) and introduce additional errors in calculating the elevation mainly due to the calculation method. The advantage of a triangle-based approach is that it does not generate any additional errors.

Figure 2 shows a part of the ASTER GDEM2, for instance. $A = (x_i^{DEM}, y_i^{DEM}, z_{ij}^{DEM})^T$ is the 2-D grid point (i, j) for which the MSL-elevation z_{ii}^{DEM} is known.

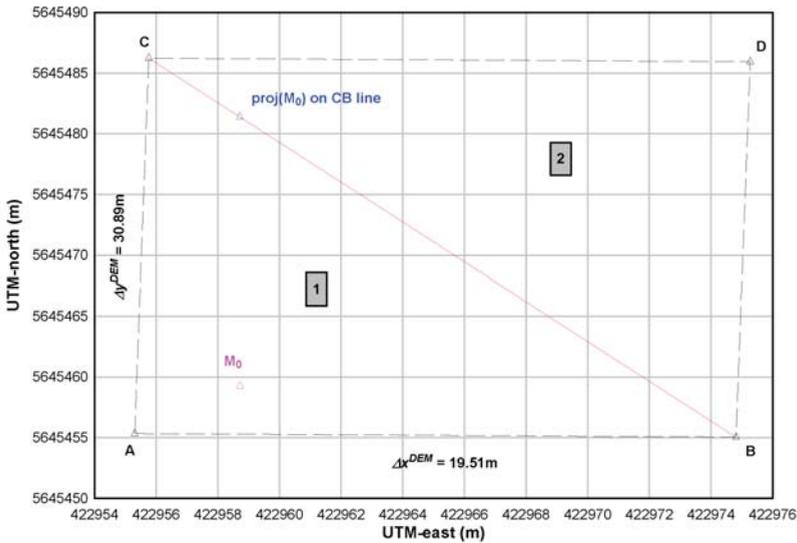


Figure 2. Example of triangulation of the ASTER GDEM2 grid.

The triangles #1 (ABC) and #2 (BCD) are defined by these additional points:

- $B = (x_i^{DEM} + \Delta x^{DEM}, y_i^{DEM}, z_{i+1,j}^{DEM})^T$
- $C = (x_i^{DEM}, y_i^{DEM} + \Delta y^{DEM}, z_{i,j+1}^{DEM})^T$
- $D = (x_i^{DEM} + \Delta x^{DEM}, y_i^{DEM} + \Delta y^{DEM}, z_{i+1,j+1}^{DEM})^T$

The fixed horizontal resolution of the grid allows to write:

$$\begin{cases} x_{i+1}^{DEM} = x_i^{DEM} + \Delta x^{DEM} \\ y_{i+1}^{DEM} = y_i^{DEM} + \Delta y^{DEM} \end{cases} \quad (29)$$

The plane equation based on the triangle #1 needs to compute the normal \vec{n}_1 to the (ABC) plane :

$$\vec{n}_1 = \vec{AB} \times \vec{AC} = \begin{pmatrix} -\Delta y^{DEM} (z_{i+1,j}^{DEM} - z_{i,j}^{DEM}) \\ -\Delta x^{DEM} (z_{i,j+1}^{DEM} - z_{i,j}^{DEM}) \\ \Delta x^{DEM} \Delta y^{DEM} \end{pmatrix} \quad (30)$$

Then, the condition for a point $M = (x, y, z)^T$ to be on the plane (ABC) is written as follows :

$$\overrightarrow{AM} \cdot \vec{n}_1 = 0 \tag{31}$$

This leads to the equation of the plane (ABC) that supports the triangle #1 :

$$z = z_{i,j}^{DEM} + \frac{z_{i+1,j}^{DEM} - z_{i,j}^{DEM}}{\Delta x^{DEM}}(x - x_i^{DEM}) + \frac{z_{i,j+1}^{DEM} - z_{i,j}^{DEM}}{\Delta y^{DEM}}(y - y_i^{DEM}) \tag{32}$$

The point A being the support point, this equation gives the vertical elevation of the point M as a function of these horizontal coordinates that is the goal to reach in the proposed method.

The point D is now the support point and the plane equation based on the triangle #2 needs to compute the normal \vec{n}_2 to the (BCD) plane :

$$\vec{n}_2 = \overrightarrow{DB} \times \overrightarrow{DC} = \begin{pmatrix} \Delta y^{DEM}(z_{i+1,j+1}^{DEM} - z_{i,j+1}^{DEM}) \\ \Delta x^{DEM}(z_{i+1,j+1}^{DEM} - z_{i+1,j}^{DEM}) \\ -\Delta x^{DEM}\Delta y^{DEM} \end{pmatrix} \tag{33}$$

The point $M = (x, y, z)^T$ belongs to the plane (BCD) if :

$$\overrightarrow{DM} \cdot \vec{n}_2 = 0 \tag{34}$$

The plane equation of (BCD) that supports the triangle #2 is :

$$z = z_{i+1,j+1}^{DEM} + \frac{z_{i+1,j+1}^{DEM} - z_{i,j+1}^{DEM}}{\Delta x^{DEM}}(x - x_{i+1}^{DEM}) + \frac{z_{i+1,j+1}^{DEM} - z_{i+1,j}^{DEM}}{\Delta y^{DEM}}(y - y_{i+1}^{DEM}) \tag{35}$$

So, Equations (32) and (35) allow to compute the vertical elevation at the point M that belongs to the triangle #1 or #2.

4.2. Identification of the Relevant Triangle and Computation of the MSL-Elevation

The challenge is now to identify the relevant triangle of the point M . At this step, the last estimator allows to fix the horizontal location of the point in the DEM by writing:

$$\begin{cases} x = x_{i+1}^{OSM} \\ y = y_{i+1}^{OSM} \end{cases} \tag{36}$$

To compute the elevation of the point M , it must be determined if the point $M_0^{DEM} = (x_0^{DEM} = x, y_0^{DEM} = y, z_0^{OSM} \neq z)^T$ belongs to the triangle (ABC) or (BCD) . For this, one computes the sum of the angles formed by the point M_0 and the tops of the relevant triangle:

$$\begin{cases} S_{(ABC)} = \widehat{AM_0B} + \widehat{BM_0C} + \widehat{CM_0A} \\ S_{(BCD)} = \widehat{BM_0C} + \widehat{CM_0D} + \widehat{DM_0B} \end{cases} \tag{37}$$

If $S_{(ABC)} = 2\pi$ the relevant triangle is #1, otherwise $S_{(BCD)} = 2\pi$ and the relevant triangle is #2. From Equations (32) or (35), the elevation z_0^{DEM} will be computed to lead to the following measurement that is used in the DEM-correction step (see part Section 3.3.3):

$$Z_0^{DEM} = M_0^{DEM} \tag{38}$$

5. Experimental Framework

A scenario has been defined to validate our method. It relies on available OSM and ASTER GDEM2 data in a selected area (3 km by 3 km) of the city of Calais, France. We focus here on the results of estimated roads inclinations.

5.1. Scenario Context

The objective is to estimate the roads inclinations. These roads are extracted from the OSM database. The experimentation context is shown in Figure 3. A car is driven on this part of OSM road network composed of motorways mainly. It is fitted with a dual-GPS system that relies on two u-blox EVK-6T receivers that collect the 3-D vehicle positionings. The receiver manufacturer announces the position accuracy as 2.5 m CEP (Circular Error Probable) and 2 m CEP if SBAS is enabled. These precisions may be more or less degraded depending on the experimental conditions (m



Figure 3. Aerial view of the OSM road network.

The GPS antennas are located on the car roof and aligned with the driver side. The spacing distance is $D = 1.75$ m. Their height from ground is $H = 1.55$ m. The improved SBAS positioning (EGNOS) has been enabled on each receiver whose GPS navigation rate has been set to 1 Hz.

The ASTER GDEM2 informs on the elevation levels of the experimentation area. They are plotted in Figure 4a,b. This dataset is composed of a grid of 14,560 MSL-elevation points that are spaced by a known horizontal distance: $\Delta x^{\text{DEM}} = \Delta y^{\text{DEM}} = 1$ arcsecond (or 30 m) in a LLH geographical datum; these values must be recomputed if a horizontal UTM datum is used. In addition, it must be noticed that some voids and artifacts (step, pit-in-bump, mole-run, etc.) remain in this version of dataset, although they announced as decreased in the northern area or almost disappeared. The ASTER GDEM2 is provided with a quality parameter that informs on the measurement accuracy of the elevation. This information can be used in the matching procedure of DEM to reject the measure, if needed.

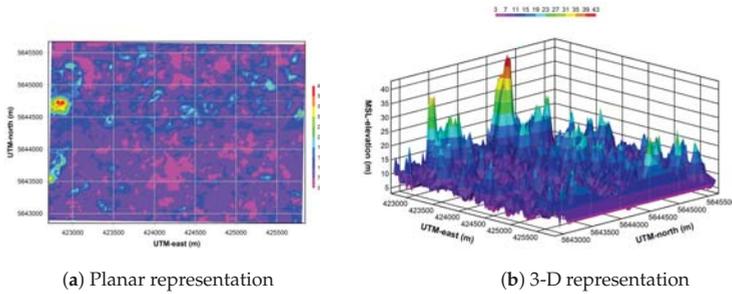


Figure 4. ASTER V2 elevation levels in the vicinity of the OSM road network.

Note that the u-blox GPS receivers directly deliver MSL-elevations in the EGM84 vertical datum while the ASTER GDEM V2 use the geoid of EGM96. The prerequisite is also to convert the u-blox MSL-elevations in the same vertical datum. This is done by using the MSP GEOTRANS software [29].

5.2. Map-Matching Procedure

Our OSM dataset is composed of 73 polylines that represent various roads as primary or secondary roads, motorways, etc. These polylines are modeled by nodes and road segments whose total number is 417 as shown in Figure 5a. The 2-D location of MSL-elevation points are also plotted (light grey) on this figure. The main objective is here to estimate the slope of each road segment on which the car is driven. Therefore, the UKF-based algorithm allows to realize the fusion of GPS/OSM/ ASTER data for estimating the car path that is plotted in black squares here. Selected enlargements of the estimated path are available in Figure 5b–d.

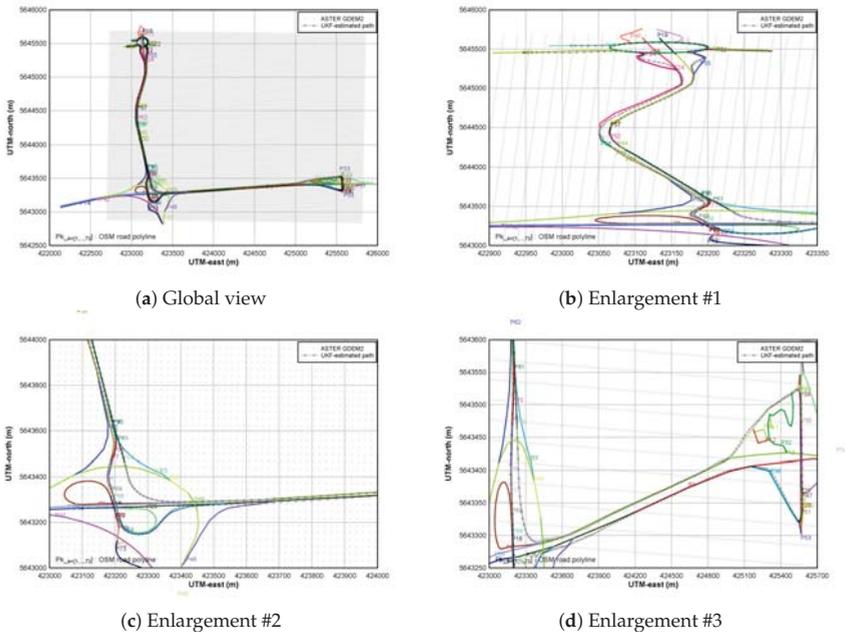


Figure 5. Estimated path of the vehicle vs. OSM road network.

Once the two GPS positioning fused, the OSM-matching step allows to identify the road segment on which the car is located. This procedure relies on the computing of Mahalanobis distances whose the time-evolution is plotted on Figure 6, highlighting ϵ that is the OSM-matching threshold. In order to refine also the ground-location of the vehicle, this threshold is used to decide when the OSM road network can be used. The map-matching step reaches a rate of 96.30%, knowing that a 100% score is difficult to get due to land surveying errors or missing roads

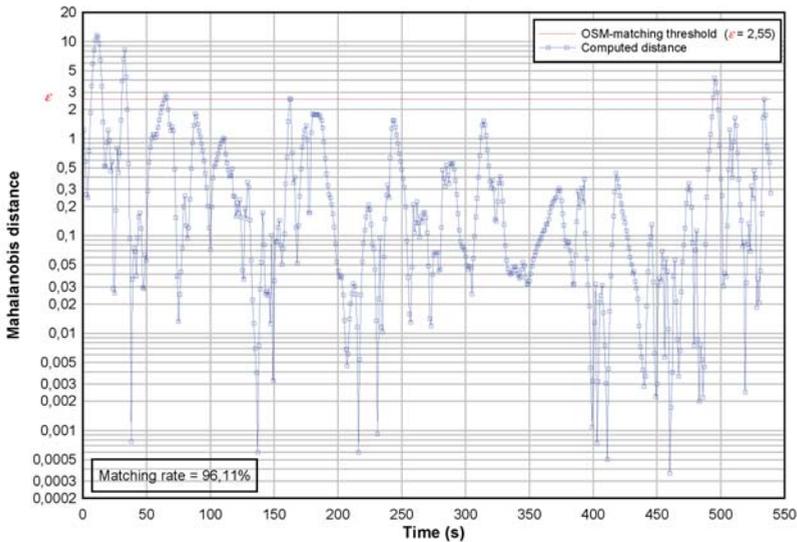


Figure 6. Time-evolution of the OSM-based matching criterion.

5.3. Road Inclination Estimation

The proposed approach fuses the GPS-based altitudes and the matched ASTER-based elevations to refine the estimated altitude of the car as shown in Figure 7 where the time-evolution is plotted in blue circles. This figure shows also the number of all matched OSM road segments—during the defined scenario—that are plotted in black squares. To estimate the slope of each road segment, the proposed least squares method needs the UKF-estimated altitudes and the matched map data as input.

As detailed in Equation (25), the road segment needs to be traveled during 2 GPS time-instants to estimate the inclination but in order to improve the estimation, the minimum number of needed GPS readings is set to 4. Figure 8 shows the results of roads inclination percentage. They concern only the OSM road segments that the car could traveled during the test scenario. Overall, 34% of traveled road segments have benefited from an estimate of their inclination. Among these estimations, about 28% of these road segments are in an area with rather flat terrain. It is possible to find an explanation in heavy climbs or descents when the vehicle has taken the entrances or exits of motorways that have undergone an elevation to their construction. The vehicle has also used exchangers that are generally on an upward or downward slope. Note that the estimation accuracy depends to the minimum number of available UKF-estimated altitudes and this parameter can be set dynamically in term of road segment length, car speed and GPS navigation rate.

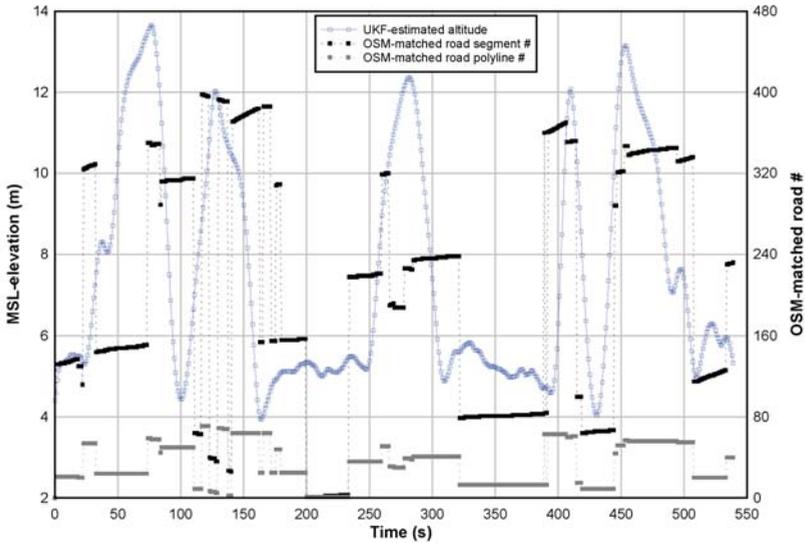


Figure 7. UKF-estimated altitudes vs. OSM matched roads.

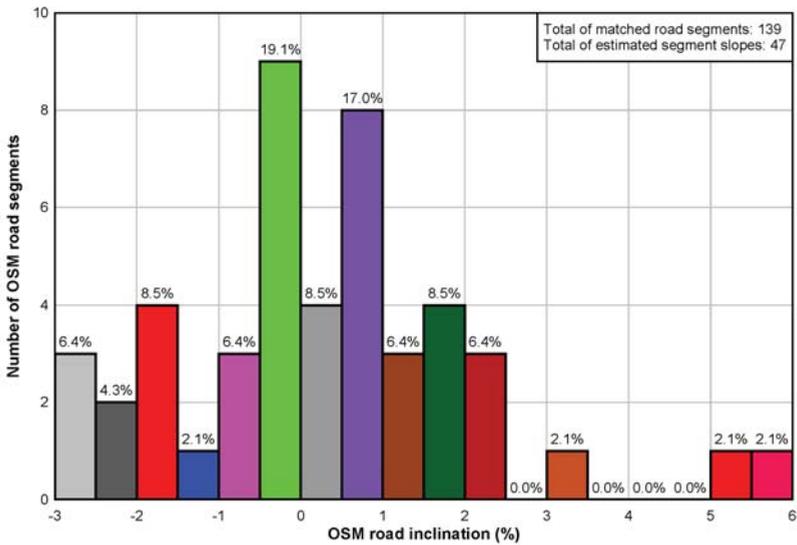


Figure 8. Estimation of the OSM road inclinations.

6. Conclusions

This work states a general approach to estimate the inclinations of OSM road segments. It relies on the collected data from a dual-GPS positioning, an OSM road network and an ASTER GDEM2 terrain surface that are fused following a centralized scheme. The proposed method takes into account the topology of the geographical area and allows a new computation of the elevation point from the terrain models. The map-matching procedure allows to select the used OSM road segments and the

triangulation of DEM leads to a refinement of the ASTER MSL-elevation. Therefore, it becomes possible to enhance the OSM database with the estimated slope of each matched road segment automatically. Then, the road inclination parameter is available for further more reliable route planning applications, for instance those of electric vehicles whose autonomy is yet limited for trips. Based on the OSM concept, our method is designed for a collaborative enhancement of the database. This general and theoretical framework is suitable to the implementation of other kinds of GNSS, DEM, etc. It provides the basis for a 3-D modeling of the road segments in a map database. Cases of GPS outages and non-existing OSM roads are not considered in this work. The next step is now to study multi-GNSS positioning systems and alternative roadmap databases.

Author Contributions: Christophe Boucher and Jean-Charles Noyer contributed equally to this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kaul, M.; Yang, B.; Jensen, C. Building Accurate 3D Spatial Networks to Enable Next Generation Intelligent Transportation Systems. In Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management (MDM), Milan, Italy, 3–6 June 2013; Volume 1, pp. 137–146.
2. Bennet, J. *OpenStreetMap*; Packt Publishing Limited: Birmingham, UK, 2010.
3. Haklay, M.; Weber, P. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Comput.* **2008**, *7*, 12–18.
4. Fathi, A.; Krumm, J. Detecting Road Intersections from GPS Traces. In Proceedings of the 6th International Conference on Geographic Information Science (GIScience), Zurich, Switzerland, 14–17 September 2010.
5. Boucher, C.; Noyer, J.C. Dual-GPS Fusion for Automatic Enhancement of Digital OSM Roadmaps. In Proceedings of the IEEE AESS European Conference on Satellite Telecommunications, Rome, Italy, 2–5 October 2012.
6. Boucher, C. Fusion of GPS, OSM and DEM Data for Estimating Road Network Elevation. In Proceedings of the IEEE International Conference on Computational Intelligence, Communication Systems and Networks, Madrid, Spain, 5–7 June 2013; pp. 273–278.
7. Tachikawa, T. *ASTER Global Digital Elevation Model Version 2—Summary of Validation Results*; Technical Report; NASA: Washington, DC, USA, 2011.
8. Farr, T.G.; Rosen, P.; Caro, E.; Crippen, R.; Duren, R.; Hensley, S.; Kobrick, M.; Paller, M.; Rodriguez, E.; Roth, L.; et al. The Shuttle Radar Topography Mission. *Rev. Geophys.* **2007**, *45*, 1–33, doi:10.1029/2005RG000183.
9. Colosimo, G.; Vendictis, L.D.; Crespi, M.; Jacobsen, K. Accuracy evaluation of SRTM and ASTER DSMs. In Proceedings of the 29th EARSeL Symposium (European Association of Remote Sensing Laboratories), Chania, Greece, 15–18 June 2009.
10. Suwandana, E.; Kawamura, K.; Sakuno, Y.; Kustiyanto, E.; Raharjo, B. Evaluation of ASTER GDEM2 in Comparison with GDEM1, SRTM DEM and Topographic-Map-Derived DEM Using Inundation Area Analysis and RTK-dGPS Data. *Remote Sens.* **2012**, *4*, 2419–2431.
11. Grip, H.F.; Insland, L.; Johansen, T.A.; Kalkkuhl, J.C.; Suissa, A. Estimation of road inclination and bank angle in automotive vehicles. In Proceedings of the American Control Conference (ACC), St. Louis, MO, USA, 10–12 June 2009; pp. 426–432.
12. Poz, A.P.D.; Gallis, R.A.B.; da Silva, J.F.C. Three-Dimensional Semiautomatic Road Extraction from a High-Resolution Aerial Image by Dynamic-Programming Optimization in the Object Space. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 796–800.
13. Wang, J.; Lawson, G.; Shen, Y. Automatic high-fidelity 3D road network modeling based on 2D (GIS) data. *Adv. Eng. Softw.* **2014**, *76*, 86–98.
14. Gong, Y.; Zhu, Y.; Yu, J. DEEL: Detecting elevation of urban roads with smartphones on wheels. In Proceedings of the 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Seattle, WA, USA, 22–25 June 2015; pp. 363–371.
15. Selim, G.; Sabour, A.A. Extending land navigation applications to support off road based on elevation and land characteristics. In Proceedings of the 2010 IEEE International Conference on Vehicular Electronics and Safety (ICVES), QingDao, China, 15–17 July 2010; pp. 65–72.

16. Mandel, C.; Laue, T. Particle filter-based position estimation in road networks using digital elevation models. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 5744–5749.
17. Chewputtanagul, P.; Jackson, D.J. A road recognition system using GPS/GIS integrated system. In Proceedings of the 2004 IEEE Region 10 Conference, TENCON 2004, Chiang Mai, Thailand, 24 November 2004; Volume D4, pp. 225–228.
18. Van der Merwe, R.; Doucet, A.; de Freitas, N.; Wan, E. *The Unscented Particle Filter*; Technical Report CUED/F-INFENG/TR 380; Cambridge University: Cambridge, UK, 2000.
19. Yao, Z.; Cui, X.; Lu, M.; Feng, Z. Dual update-rate carrier tracking technique for new generation global navigation satellite system signals in dynamic environments. *IET Radar Sonar Navig.* **2009**, *3*, 203–213.
20. Cabani, A.; Khemmar, R.; Ertaud, J.Y.; Mouzna, J. Intelligent navigation system-based optimization of the energy consumption. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 785–789.
21. Julier, S.; Uhlmann, J. A New Extension of the Kalman Filter to Nonlinear Systems. In Proceedings of the SPIE Aerosense Symposium, Orlando, FL, USA, 20–25 April 1997.
22. Arulampalam, M.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188.
23. Julier, S.J.; Uhlmann, J.K. *A General Method for Approximating Nonlinear Transformation of Probability Distribution*; Technical Report; University of Oxford: Oxford, UK, 1996.
24. Quddus, M.A.; Ochieng, W.Y.; Noland, R.B. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transp. Res. Part C* **2007**, *15*, 312–328.
25. Mahalanobis, P.C. On the Generalised Distance in Statistics. In *Proceedings of the National Institute of Sciences of India*; Springer: Berlin, Germany, 1936; Volume 2, pp. 49–55.
26. Boucher, C.; Noyer, J.C. Automatic estimation of road inclinations by fusing GPS readings with OSM and ASTER GDEM2 data. In Proceedings of the IEEE International Conference on Connected Vehicles & Expo, Vienna, Austria, 3–7 November 2014; pp. 871–876.
27. Tighe, M.L.; Chamberlain, D. *Accuracy Comparison of the SRTM, ASTER, NED, NEXTMAP USA Digital Terrain Model over Several USA Study Sites*; Technical Report; Intermap Technologies: Englewood, CO, USA, 2009.
28. Kaplan, E. (Ed.) *Understanding GPS: Principles and Applications*; Artech House: Boston, MA, USA, 1996.
29. BAE Systems. *User's Guide for MSP Geographic Translator (GEOTRANS) Version 3.5*; BAE Systems: San Diego, CA, USA, 2014.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

MDPI AG

St. Alban-Anlage 66
4052 Basel, Switzerland
Tel. +41 61 683 77 34
Fax +41 61 302 89 18
<http://www.mdpi.com>

Sensors Editorial Office

E-mail: sensors@mdpi.com
<http://www.mdpi.com/journal/sensors>



MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland

Tel: +41 61 683 77 34
Fax: +41 61 302 89 18

www.mdpi.com



ISBN 978-3-03842-934-0