Article

Recognition and Location Algorithm for Pallets in Warehouses Using RGB-D Sensor

Junhong Zhao [1,2]**, Bin Li** [1,2,]*****, Xinyu Wei** [1,2]**, Huazhong Lu** [2,3]**, Enli Lü** [4] **and Xingxing Zhou** [1,2,]*****

[1] Institute of Facility Agriculture, Guangdong Academy of Agricultural Sciences, Guangzhou 510640, China
[2] Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou 510642, China
[3] Guangdong Academy of Agricultural Sciences, Guangzhou 510640, China
[4] Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education,
   South China Agricultural University, Guangzhou 510642, China
***** Correspondence: ganli180@sina.com (B.L.); zhouxingxing@gdaas.cn (X.Z.); Tel.: +86-020-3881-5805 (X.Z.)

Featured Application: A real-time approach based on an improved label template matching algorithm with an RGB-D sensor was proposed to recognize and locate pallets in a warehouse environment in this paper. The application of this technology can reduce labor and cost savings, as well as provide a level of warehouse automation.

Abstract: (1) Background: Forklifts are used widely in factories, but it shows the problem of large uncertainties when using an RGB-D sensor to recognize and locate pallets in warehouse environments. To enhance the flexibility of current autonomous forklifts in unstructured environments, the improved labeled template matching algorithm was proposed to recognize pallets. (2) Methods: The algorithm comprises four steps: (i) classifying each pixel of a color image with the color feature and obtaining the category matrix; (ii) building a labeled template containing the goods, pallet, and ground category information; (iii) compressing and matching the category matrix and template to determine the region of the pallet; and (iv) extracting the pallet pose from information in respect of the pallet feet. (3) Results: The results show that the proposed algorithm is robust against environmental influences and obstacles and that it can precisely recognize and segment multiple pallets in a warehouse with a 92.6% detection rate. The time consumptions were 72.44, 85.45, 117.63, and 182.84 ms for detection distances of 1000, 2000, 3000, and 4000 mm, respectively. (4) Conclusions: Both static and dynamic experiments were conducted, and the results demonstrate that the detection accuracy is directly related to the detection angle and distance.

Keywords: pallet recognition; autonomous forklift; labeled template; RGB-D sensor

1. Introduction

Forklifts are one of the most popular types of handling equipment on the market [1], and the demand for them tends to increase year by year [2], because they offer many benefits, such as improved productivity and reduced manual handling [3]. The Research and Markets research agency predicts that the forklift market will record sales of about 2.2 million units in 2023, with a compound annual growth rate of about 9%. According to a report by the PR Newswire agency, the forklift market will grow at a rate of 7.8% annually, increasing from 2 billion USD in 2020 USD to 2.9 billion in 2035 [4]. The application of such a large number of forklifts has and will result in their involvement with high accident and fatality rates around the world [5–7]. Researchers have reported that most accidents can be attributed to human operator errors, such as a lack of attention, misperception, or misjudgment [8,9]. The US Occupational Safety and Health Administration (OSHA) also reported that a significant number of forklift accident cases are caused by

reduced situation awareness of the forklift operator [10,11]. The prevalence of this type of accident clearly highlights the importance of developing automatic-drive forklifts.

The use of automatic-drive forklifts still requires the development of key technologies for tasks such as autonomous navigation, pallet location, and pallet recognition. Automated guided vehicles (AGVs) have been proposed since the early 1950s, and the first AGV was created in America [12,13] to help improve the autonomous navigation of forklifts. Researchers have intensely studied optimal control theories and optimization methods for solving AGV path planning and navigation problems in recent years [14–17]. Zhang et al. [15] proposed a learning-based algorithm to solve global path planning problems, which involved training a deep convolutional neural network with dual branches (DB-CNN). However, it is more difficult to recognize and locate pallets in actual warehouses because pallets are placed with high uncertainty. To address this issue, the authors of [18,19] defined a semi-structured environment representing work conditions that contained information about pallets and the environment as a priori information, including the desired pose, goods loaded on pallets, and goods on a shelf or the ground.

The present investigation aimed to address three key issues that must be overcome in semi-structured environments. First, a forklift algorithm has to identify pallets with uncertainty and subsequently maintain a continuously accurate estimation of the pose parameters while approaching and picking up pallets [19]. The pose of a target pallet cannot be predetermined due to errors in human operation. Second, a forklift algorithm should identify various pallets because several types of pallets and goods are used in warehouses. Third, the algorithm should have the ability to segment multiple pallets. Segmenting pallets situated within a small distance of each other is difficult. If the pallet detection algorithm cannot overcome these problems, then the fork may damage goods and lead to security incidents.

Some researchers have explored various algorithms for pallet recognition based on a single sensor. An algorithm based on calculating geometry using pallet edges and shape features was investigated, but the features could be incorrectly extracted when the pallet was blocked or the illumination conditions changed. In [19,20], the pose of single pallets was estimated based on geometry features extracted from distance data provided by LiDAR with geometry classifiers. In addition, the authors of [21–26] obtained pallet pose information based on pallet size and edge features extracted from a color image provided by a camera. To reduce the effect of the environment, pallets were identified with marks attached to the pallet feet in [27–29]. However, the processes of adding and calibrating the mark were found to lead to higher costs and more human work.

Additionally, some researchers have focused on recognizing pallets from images and extracting pose parameters using depth sensors such as laser radar and structured light. The authors of these studies were able to achieve more robust performance in semi-structured environments. In [18], a three-foot pallet template was created to match the scattering image of LiDAR data. In a color image, an edge template was used to match the distance-transform image. The two results were combined based on the maximum percentage of couple points. In [30], a pallet model was created and pose was estimated using an ICP method applied to LiDAR data. In [31], a classifier with a feature grid template and gray-scale features (normalized pair differences) was proposed, and it was shown that pallets could be identified with the classifier. The algorithms used color thresholds in [32–34]. Moreover, other researchers trained a classifier [35] to detect a wooden pallet from images using Haar-like features, and the results were verified with an adaptive structural feature and direction-weighted overlapping ratio in [36].

The aforementioned approaches used different types of sensing, but they all involved single-sensor devices, such as laser scanners and cameras (monocular or stereo). A laser scanner only provides distance information, but it has the advantages of being stable and robust against light. In contrast, a camera provides vision data that include rich information. However, an RGB-D sensor combines a camera and a distance range sensor to obtain both color and distance information. The camera obtains image data by CMOS, and

each pixel of the image records red, green, and blue color data [37–40] that are used for recognition. The depth sensor obtains depth information for every pixel. For tray image recognition, researchers have mostly focused on the recognition of a single tray at the expense of the spatial relationship between trays and the environment. For pallet image recognition, researchers have mostly focused on the recognition of a single pallet at the expense of the spatial relationship between pallets and the environment [41–43].

This paper proposes an algorithm to recognize pallets and locate them using an RGB-D sensor in a semi-structured environment. This algorithm was adapted to situations with multiple pallets and with differently shaped pallets. An accuracy experiment was conducted to test the algorithm. A labeled template matching algorithm was developed to accelerate the matching speed, and this algorithm served as a basis for studying object detection in unstructured environments. The simplicity of operation should enhance the flexibility of autonomous forklifts and expand the applicability of autonomous forklifts in complex environmental conditions.

2. Methods

2.1. Establishing Forklift Detection Model and Situation Analysis

Figure 1 shows the considered situation detection model of approaching a pallet. The model consists of a forklift, RGB-D sensor, goods, and pallets. Multiple pallets are used to load differently sized goods, and the pallets are located on the ground, goods, or shelves. Two coordinate frames were constructed: world coordinate frame {W}, where the X–Y plane is the ground, and sensor coordinate frame {C}, where the X–Y plane is parallel to the ground. The detection algorithm aims to obtain the center position of the pallet surface in {C} and the angle of {C} to Y–Z, which is represented as ($^{c}x_p$, $^{c}y_p$, $^{c}z_p$, θ). The pallet has four degrees of freedom: moving along the X, Y, and Z axes of {W} and rotating along the Z axis of {W} (Figure 1).

In previous studies, scholars attempted to recognize single pallets without taking the goods, the ground, and shelves into account. However, not only pallets but also goods and the ground must be detected during the process of engaging a pallet. Moreover, the combination of pallets and goods must always be predetermined such that information in respect of goods and the ground can be utilized as the detection target. With more target information, a detection algorithm can be more robust in complicated situations.

(a)

(b)
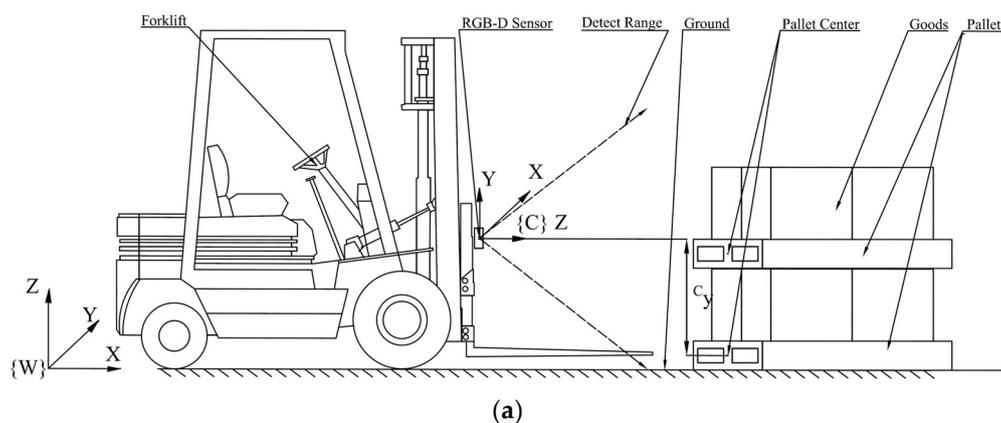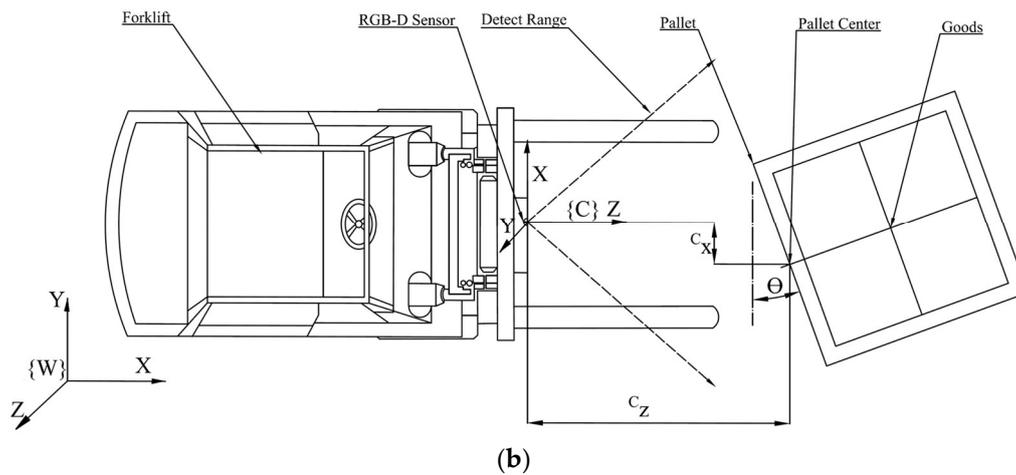
Figure 1. Schematic of forklift detection situation. (**a**) Side view schematic. (**b**) Overhead view schematic

2.2. Algorithm Flow

The algorithm is divided into three steps: calculating initial data, obtaining the range of interest (ROI) of the pallet, and estimating pallet pose parameters. After running step 1 once, the system loops through steps 2 and 3 to update the pose parameters of the pallet, as shown in Figure 2.

In step 1, the algorithm obtains information in respect of pallets and goods and then selects the classifier model that classifies each pixel of the color image. The initial distance is a statistic from the distance data of pallet category pixels, which is the initial parameter for the next step.

In step 2, the algorithm obtains a new RGB and depth image from the sensors. If the target distance is in the effective range, then the compression unit size is calculated based on the projective principle. After classifying each pixel, a category matrix records the label information. A template is created based on the target information. To accelerate the matching process, both the category matrix and template are compressed. The labeled template is matched to the category matrix, and the match score of pixels is calculated. If the match score is higher than the threshold score, then the pallet ROI is confirmed.

In step 3, the foot coordinates of the pallet are obtained from the ROI using the pallet geometry information. In sequence, the pallet center coordinate and the pallet angle are calculated. A sliding average filter is constructed to filter the pose parameters. Finally, the distance of the target is updated to step 2 to calculate the template size.

This algorithm achieves real-time parameter updates. All the details are described in following sections.
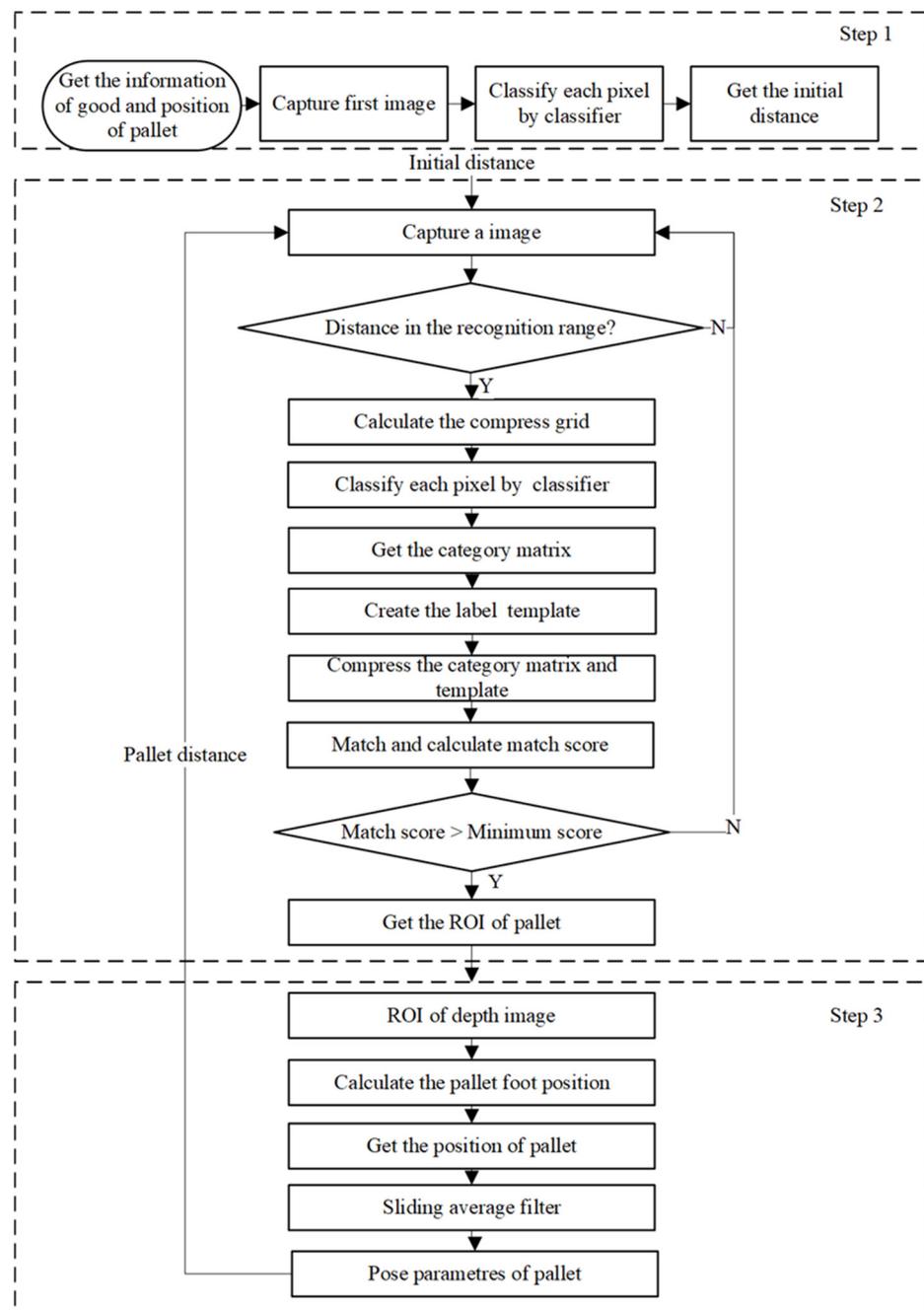
Figure 2. Flow chart of the proposed algorithm.

2.3. Classifier Training

2.3.1. Analysis and Feature Selection

The choice of features has a considerable impact on the speed and accuracy of an algorithm. Due to real-time and pixel-level accuracy requirements, the features should be easily extracted, pixel-level, and distinguishable. Three different types of pallets are presented in Figure 3. As mentioned above, the objects to be recognized are pallets, goods, and the ground/shelf. The most common features can be divided into three parts: shape, texture-based features, and color. The features are next discussed with reference to the real situation.

(1) It is difficult to apply the same shape feature to represent all pallets and quickly extract information because a pallet's shape and size are variable from changing angles.

(2) Texture-based features are generally local features that use a statistical vector to express the gray change in the cell of each pixel, such as histogram of oriented gradient (HOG), local binary patterns (LBP), and scale-invariant feature transform (SIFT). These features are widely used in complex object recognition tasks [44,45]. However, pallets are often composed of plastic with a smooth and flat surface. Occasionally, texture information may not be enough for the algorithm to segment the target and background, and more time is needed for classification due to expensive computation.

(3) Because a source image is recorded in the RGB channel format, it can be transformed into HSI color space to help easily segment a target with little computational cost. As shown in Figure 2, the color feature can fit most situations. Therefore, effective color components R, G, B, H, and S were selected as the input features to train the classifiers in this study.

| (**a**) | (**b**) | (**c**) |

Figure 3. Pallet examples in warehouse. (**a**) Example a. (**b**) Example b. (**c**) Example c.

2.3.2. Classifier Construction and Category Matrix Creation

This study uses a data set with the aim of performing pixel classification. The data set contains 21 RGB images of three types of pallet taken at different distances. The dataset was collected and photographed with a Bumblebee XB3 camera (supplied by the Point Grey company) in the XiZhou agriculture material warehouse, XinTang, Guangdong. The dataset was annotated with Photoshop (Adobe Systems Incorporated). The pixels corresponding to goods, pallet, pallet hole, and the ground are represented as A, B, C, and D, respectively. Each category contains more than 20,000 pixels as the classifier sample and the sample size of each category is balanced. This study divides the training and testing into 80% and 20%, respectively.

The input vector x is presented as x_{ij}, which contains *R*, *G*, *B*, *H*, and *S* color information of pixels, which is shown in Equation (1). The output is the category corresponding to each pixel, which is represented by numbers, as shown in Equation (2).

$$x_{ij} = \begin{bmatrix} R \\ G \\ B \\ H \\ S \end{bmatrix} \tag{1}$$

where *i* and *j* represent the pixel's column and row, respectively.

$$c_i = \begin{cases} 1 & x_i \in A \\ 2 & x_i \in B \\ 3 & x_i \in C \\ 4 & x_i \in D \end{cases} \tag{2}$$

This study uses support vector machine (SVM) as the pixel classifier. SVM classifier is a supervised machine learning technique, and it is one of the most popular discriminative classifiers [46]. This classifier applies the kernel trick to maximum-margin

hyperplanes to solve linear and nonlinear classification problems. A linear discriminant function is defined as shown in Equation (3).

$$f(x) = sgn(w^T * x + b^*) \tag{3}$$

where w^T and b^* are the optimal parameters of maximum-margin hyperplanes, which were obtained from Equation (4).

$$arg \min_{w,b} \frac{1}{2} \|w\|^2 \tag{4}$$

$$subject\ to\ \ y_{ij}(w^T x_{ij} + b) \geq 1, \forall i, j$$

The classifier was trained with an SVM model from sklearn, which is an open source library.

Different pallet loading scenarios are trained with corresponding different classification models. In the standardized warehouse, goods and pallets correspond to each other, and different pallets are generally used for different goods, as shown in Figure 3, so the target goods can be used as a priori information to determine the type of pallets selected and choose the corresponding classification model.

The RGB images acquired by the RGB-D sensor are input to the SVM classification model to obtain the category corresponding to each pixel point. The categories are stored into the category matrix to realize the category information and spatial relationship storage. Figure 4 shows the result of the classification of Figure 3a into a category matrix, and the colors of different categories are the same as the templates.
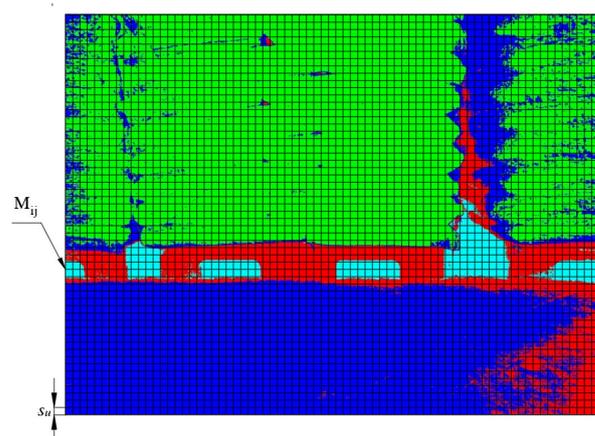
Figure 4. Category matrix example.

Element labels are represented with different colors. Green represents goods. Red represents pallets. Blue represents ground. Light blue represents pallet holes. s_u and M_{ij} represent the grid compressing unit size and the grid unit vector of category matrix, respectively (explained in Section 2.5 below).

2.4. Labeled Template Creation

As shown in Figure 3, in a warehouse environment, pallets have the same spatial characteristics: goods are placed above the pallets, the pallet is set on the ground, and the pallets correspond to the category of the goods. Therefore, we have established a labeled template that preserves the dimensions of the pallet and the spatial characteristics mentioned above, as shown in Figure 5. The labeled template contains 4 categories: goods, pallet, pallet hole and the ground or shelf. The height of the goods and the ground label of the template are set as the height of the pallet.
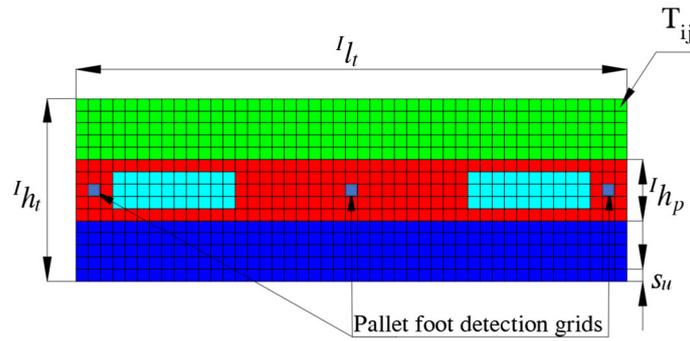
Figure 5. Labeled template example.

Element labels are represented with different colors. Green represents goods. Red represents pallets. Blue represents ground. Light blue represents pallet holes. The grids are compression units of label templet. s_u and T_{ij} represent the grid compressing unit size and the grid unit vector of Labeled templet, respectively (explained in Section 2.5 below). The pallet foot detection grids are explained in Section 2.7 below.

Based on the projection principle of the camera and the spatial relationship, as shown in Figure 6, the pixel size of the pallet in the image coordinate system {I} is calculated with Equations (5) and (6), when the distance between the pallet and the camera is $^{C}z_p$. The whole template size is calculated with Equations (7) and (8).
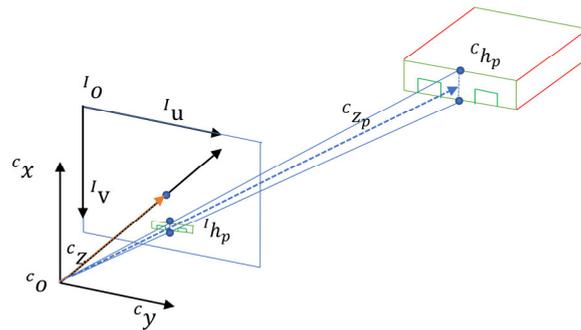
Figure 6. Diagram of the spatial relationship between the sensor and the pallet.

$$^{I}h_p = Round(\frac{f \cdot {}^{C}h_p}{{}^{C}z_p \cdot dy}) \tag{5}$$

$$^{I}l_p = Round({}^{I}h_p \frac{{}^{C}l_p}{{}^{C}h_p}) \tag{6}$$

$$^{I}h_t = 3 \times {}^{I}h_p \tag{7}$$

$$^{I}l_t = {}^{I}l_p \tag{8}$$

where $^{C}h_p$ and $^{C}l_p$ represent height and length of pallet in camera coordinate system {C}, respectively. $^{I}h_p$ and $^{I}l_p$ represent the number of pixels of the pallet height and length in image, respectively. f represents the focal length of the RGB camera in the RGB-D sensor. dy represents the pixel size of the CMOS in the RGB camera.

2.5. Grid Compressing Template and Category Matrix

Because the size of the template and category matrix has a great impact on the speed of the algorithm, we propose a grid compression algorithm to speed up the algorithm by compressing the template and category matrix with less information loss. Because the size of the template and category matrix has a great impact on the speed of the algorithm, we propose a grid compression algorithm that compresses the template and category matrix

to speed up the algorithm [47]. Meanwhile, the grid compression algorithm improves the robustness of the algorithm and reduces the impact of misclassification of pixels (shown in Figure 4) on the matching process. Because there are only four categories and most of the regions in them represent the same category, the template and category matrices contain a lot of redundant information, so the matrix size can be reduced, and most of the information can be preserved by grid compression.

The grid compression operation is shown in Figures 4 and 5. The template and category matrix are compressed in the same proportion based on the grid. The size of each compression unit is $s_u \times s_u$. s_u is calculated based on the size of the pallet in the template. The calculation principle is that the compressed grid of the template has the same category, and it retains the pallet structure information in the template. After compression, the template and category matrix are s_u times smaller. Finally, the information on the proportional number of categories in the compressed grid unit is retained by means of the homogenization vector. The grid units of the template and category matrices are represented by T_{ij} and M_{ij}, respectively, as in Equations (9) and (10).

$$T_{ij} = [\frac{c_1}{\sum_{n=1}^{4} c_n} \quad \frac{c_2}{\sum_{n=1}^{4} c_n} \quad \frac{c_3}{\sum_{n=1}^{4} c_n} \quad \frac{c_4}{\sum_{n=1}^{4} c_n}] \tag{9}$$

$$M_{ij} = \begin{bmatrix} \frac{c_1}{\sum_{n=1}^{4} c_n} \\ \frac{c_2}{\sum_{n=1}^{4} c_n} \\ \frac{c_3}{\sum_{n=1}^{4} c_n} \\ \frac{c_4}{\sum_{n=1}^{4} c_n} \end{bmatrix} \tag{10}$$

where c_n represents the number of pixels of category n in the T_{ij}, M_{ij} grid unit.

2.6. Template Matching

The position of the pallet was set to be determined by matching the template and category matrix with the sliding window method [48,49]. The matching process was a convolution operation. The template matrix matches the category matrix to calculate the matching degree. The matching degree is evaluated by the matching score of grid units, and each unit's score is calculated based on Equation (11). A higher matching score corresponds to a higher probability of being a pallet. If the matching score is higher than MiniScore, the pallet position is determined. MiniScore is determined with a threshold, as shown in Equation (12). Multiple pallets can reach the peak of the matching score matrix, and pallets have no overlap. Thus, a non-maximum suppression algorithm is used to obtain multiple pallet coordination.

$$S_{ij} = \sum_{m=0,n=0}^{H_t,L_t} T_{mn} M_{(i+m)(j+n)} \tag{11}$$

$$MiniScore = s_{thred} \, {}^{I}h_p \cdot {}^{I}l_p \tag{12}$$

where S_{ij} is the matching score of each unit, and s_{thred} is the threshold of the matching score.

2.7. Pose Parameters of Pallet Estimation

The pallet pose parameters include the center coordinates of the pallet detection surface ${}^{C}P$ and the inclination angle θ of the pallet to be measured, which is shown in Figure 1. These parameters are calculated from the depth information of the pallet foot, which is extracted from the RGB-D camera depth data. In order to reduce the influence of the angle between the sensor and the pallet, the center grids of the pallet feet are selected as the detection sampling area, as shown in Figure 1.

In the camera coordinate system {C}, the calculation formula of the pallet pose parameter is shown as Equations (13) and (14), respectively. $^C P$ is calculated by averaging the corresponding point cloud data in the detection grid of the center pallet feet. θ is calculated by fitting the slope of all point clouds within the detection grid of the pallet feet in the x-y plane of the camera coordinate system {C}. The average filter is used to smooth the successive detection results, and the estimated value of the results will be updated every cycle. The calculation formula is shown in Equations (15) and (16). Note that when dynamically detecting pallets, the detection results of each loop will be transformed to world coordinates before estimation.

$$^C P = [\, ^C \bar{x}_p \quad ^C \bar{y}_p \quad ^C \bar{z}_p \,] \tag{13}$$

$$\theta = \arctan\left(\frac{N \sum {}^C x_n \, {}^C z_n - \sum {}^C x_n \sum {}^C z_n}{N \sum {}^C x_n{}^2 - (\sum {}^C x_n)^2}\right) \tag{14}$$

$$^C \hat{P} = \frac{{}^C P^N + {}^C P^{N-1} + \cdots + {}^C P^1}{N} \tag{15}$$

$$\hat{\theta} = \frac{\theta^N + \theta^{N-1} + \cdots + \theta^1}{N} \tag{16}$$

$^C P^N$ and θ^N represent the estimated values obtained for the Nth detection. $^C \hat{P}$ and $\hat{\theta}$ represent the estimated value of $^C P$ and θ after filtering, respectively.

3. Algorithm Performance Test

Four experiments were designed to test the performance of the algorithm: (1) algorithm comparison, (2) multiple pallet recognition, (3) accuracy test in static experiment, and (4) accuracy test in dynamic experiment.

3.1. Template Matching Algorithm Comparison

To demonstrate the advantages of the proposed algorithm, a comparison was performed with the template matching algorithms proven by OpenCV 3.0. Those algorithms were tested with the labeled template without compressing the template and category matrix. Gray template and color template tests were not performed because of their low success rates in changing environments. Three typical situations were tested: large angle, background changes, and obstacle obstruction. The figures below present comparisons of these situations.

3.2. Multiple Pallet Recognition Algorithm Comparison

To illustrate the performance of the proposed recognition algorithm in recognizing multiple and different pallets in a warehouse environment where different goods are stored, the proposed algorithm is compared with the deep learning-based yolov5 approach in this paper.

A pallet dataset was photographed with a Bumblebee XB3 camera in the XiZhou agriculture material warehouse (located in XinTang, Guangdong, China). The test dataset contained 21 pallet pictures with different distances and arrangements. The training set contains 122 images captured in the warehouse.

Since the dataset is too small, yolov5 uses mosaic data enhancement, which is stitched with 4 images, randomly scaled, randomly cropped, and randomly lined up, to enrich the dataset. It also uses a pre-trained model to speed up the training speed. yolov5 takes the input images through the focus structure, downsamples the images, and then sends them to the backbone. The backbone uses CSPDarknet53, and resizes all the input images into 640 × 640 size, and the training batch size is 64, with 300 epochs of training, a learning rate of 0.01, a decay of 0.0005, and a momentum of 0.937.

To verify the performance of the proposed method, precision and recall are used as evaluation metrics in this study.

Precision indicates the proportion of the correct detection results in the total detection results, and recall indicates the proportion of the correct detection results in all the ground truths. They can be denoted as the following equations:

$$recision \ = \ \frac{TP}{TP + FP} \tag{17}$$

$$Recall \ = \ \frac{TP}{TP + FN} \tag{18}$$

where *TP*, *FP*, and *FN* represent the number of correctly detected objects, falsely detected objects, and missed detected objects, respectively.

3.3. Accuracy Test in Static Experiment

Accuracy and time consumption are important indicators for a pallet recognition algorithm. An experiment was designed to test the relationships between detection accuracy, pallet distance, and angle. This experiment was conducted in a room to easily control the environmental conditions. The sensor and pallet were set as shown in Figure 1. The pallet was moved to change the distance ^{c}z and the angle θ after fixing the sensor. The result was transformed into {W} space with the pose of the sensor.

The distance ^{c}z values were chosen as 1000, 2000, 3000, and 4000 mm because the detection range of the depth sensor is 500–4500 mm. θ was chosen as 0, ±5°, ±10°, ±15°, ±20°, and ±25°. The estimation results and errors obtained by the algorithm were recorded and counted. The overall computation time and the computation time of the main processes of the algorithm—classifying the source image, creating the template, compressing, matching, extracting ROI, and estimating pallet parameters—were also recorded.

The proposed algorithm was compiled based on OpenCV 3.0, running on a PC with 16 GB of RAM, an Intel Core i7-6820 processor, and a Windows 10 operating system. The sensor was a Kinect 2 (Microsoft, America), which can obtain RGB images with a resolution of 1920 × 1080 pixel and depth images with a resolution of 512 × 424. Kinect 2 detects the distance based on the time of flight, the depth detection range is 0.5–4.5 m and detection accuracy is ±4 mm. The depth image size was 512 × 424. The pallet size was 1300 × 1300 × 150 mm.

3.4. Accuracy Test in Dynamic Experiment

In this paper, dynamic performance refers to the algorithm's measurement accuracy when the RGB-D sensor was approaching the pallet. In this experiment, the RGB-D sensor moved close to the pallet with a changeable distance and angle between the pallet and sensor to show the influence of changing distance and angle on the engaging process. To concisely show the parameters of the experiment, all results data were transformed into {W} space, as shown in Figure 1. The detection result was transformed from {C} space, with the RGB-D sensor position and angle in {W} provided by location sensor NAV350 (SICK, Germany, scanning frequency: 8 Hz, positioning accuracy ±4 mm), a laser scanner sensor that could obtain the position and angle of the RGB-D sensor by detecting reflection marks fixed in the room.

The position of the pallet was fixed at (0, 0, 75), and θ was 90°. The sensor moved in from 4 m to 1.5 m and changed angle in three different ranges: 0–10°, 20–30°, and 0–40°. The pose of the sensor and the error of each parameter were recorded.

4. Results

4.1. Template Matching Algorithm Comparison

The results of the experiment are shown in Figure 7. Both the OpenCV algorithm and the proposed algorithm recognized the pallet when the angle between the sensor and

pallet was in the range 0–10°. However, as the angle increased, the OpenCV algorithm occasionally failed to detect the pallet. When the angle was larger than 15°, the OpenCV algorithm drifted and lost the target but the proposed algorithm was able to identify the pallet (see Figure 7a,d). Moreover, the OpenCV template matching algorithm could have been influenced by the background and blocked by obstacles, as shown in Figure 7b,e. The proposed algorithm was able to resist the impact of the environment and change in pallet, as shown in Figure 7c,f.
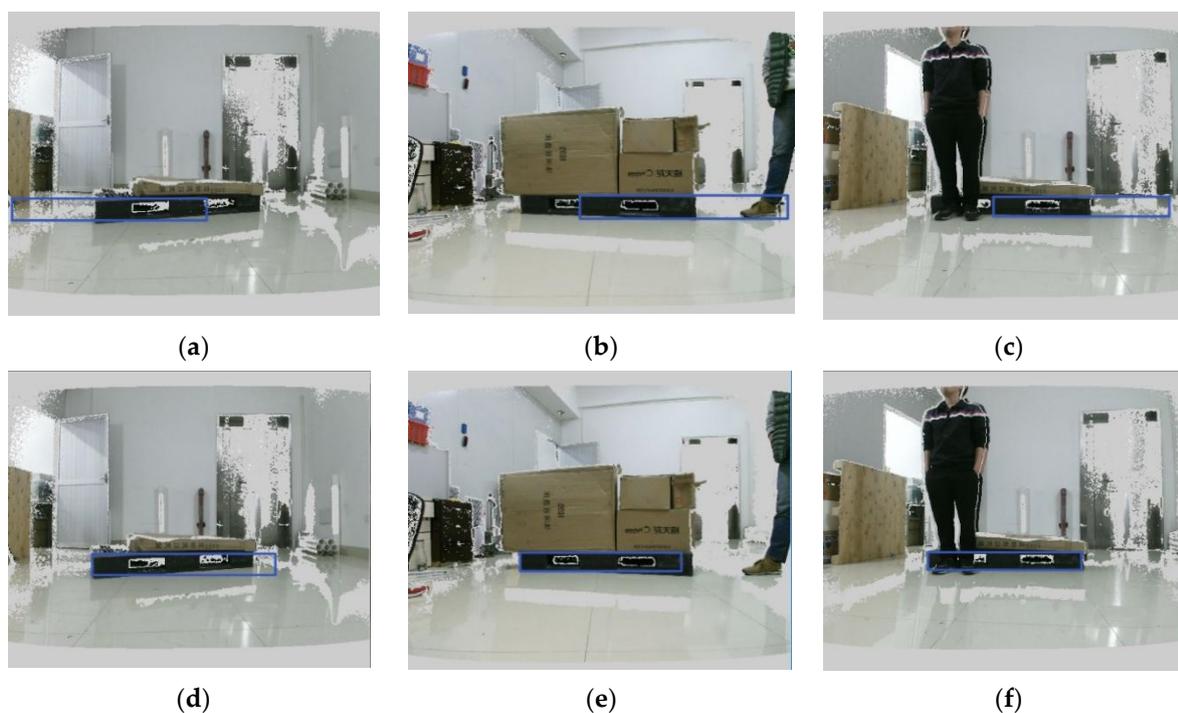
Figure 7. Comparison of experimental results under typical situations. The blue box represents the results of the recognition algorithm. (**a**) Recognition result with the template matching algorithm provided by OpenCV when the pallet angle was 15°. (**b**) Recognition result with the template matching algorithm provided by OpenCV when the pallet was subject to environmental influence. (**c**) Recognition result with template matching algorithm provided by OpenCV when the pallet was blocked by obstacles. (**d**) Recognition result with the proposed algorithm when the pallet angle was 15°. (**e**) Recognition result with the proposed algorithm when the pallet was subject to background disturbance. (**f**) Recognition result with the proposed algorithm when the pallet was blocked by obstacles.

4.2. Multiple Pallet Recognition

The multiple pallet recognition results are shown in Figure 8, where red rectangles indicate the ROI of pallets marked by the algorithm. The precision and recall of the proposed method in this paper are 0.989, 0.926. Seventy-five pallets were recognized from eighty-one pallets. The precision and recall of the yolo v5 are 0.964, 0.986.

(a)　　　　　　　　　　　　　　(b)

(c)　　　　　　　　　　　　　　(d)
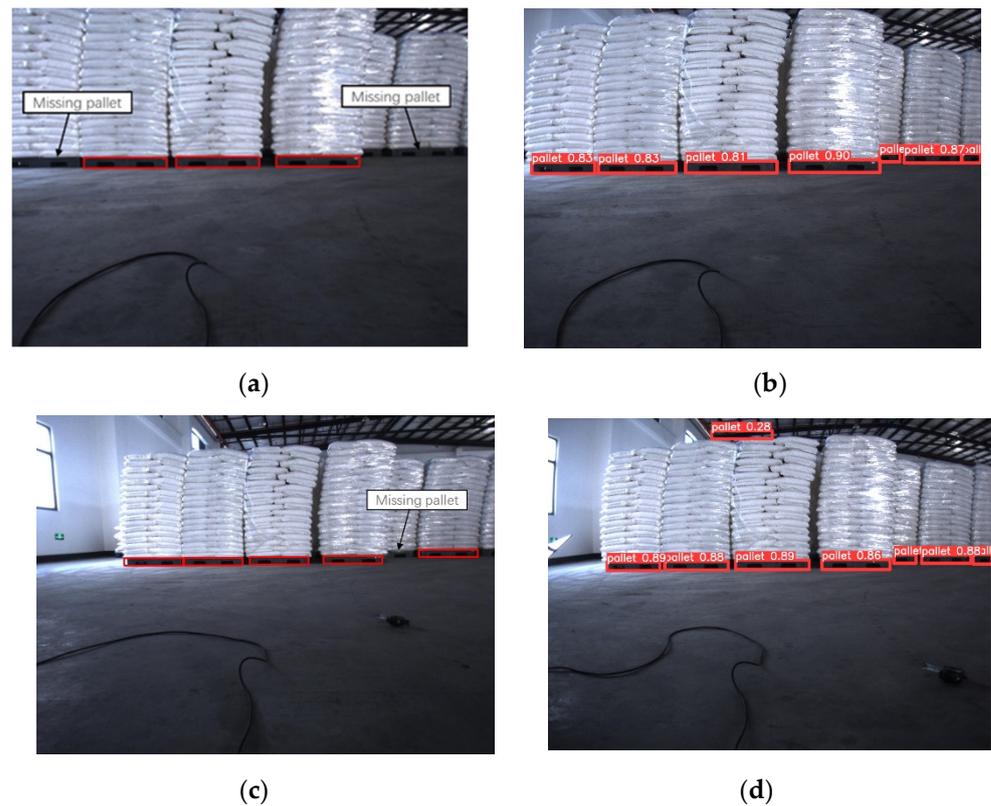
Figure 8. Multiple pallet recognition results contrast: The red box represents the results of the recognition algorithm. (**a**,**c**) is the recognition result of the proposed method, and (**b**,**d**) is the recognition result of yolov5.

4.3. Accuracy Test in Static Experiment

The estimation errors are shown in Figure 9. The curves were calculated based on the results of performing detection 50 times. Ex, Ey, Ez, and Eangle represent the errors of ^wx, ^wy, ^wz, and θ, respectively. The error was larger with a longer detection distance. When the distance was less than 2000 mm, the algorithm showed better and more stable performance. The error was larger when the pallet angle and distance increased. When the angle was larger than 20°, the standard deviation (STD) of each error increased. The maximum values of Ex, Ey, Ez, and Eangle were 94.64 mm, 44.74 mm, 63.11 mm, and 6.07 mm, respectively. The correlation coefficient (COF) between the error and angle of the pallet is shown in Table 1. The COF represents the correlation between two variables. The error had a positive correlation to the angle of the pallet, except for Ex, Ey, Ez, and Eangle at 3000 mm and for Ey and Eangle at 4000 mm.

The time consumptions of the proposed algorithm were 72.44, 85.45, 117.63, and 182.84 ms and the STDs were 0.81, 0.59, 14.45, and 13.02 at 1000, 2000, 3000, and 4000 mm, respectively. Less time was consumed when the distance was smaller. Moreover, the computation time had no relation to the angle of the pallet.
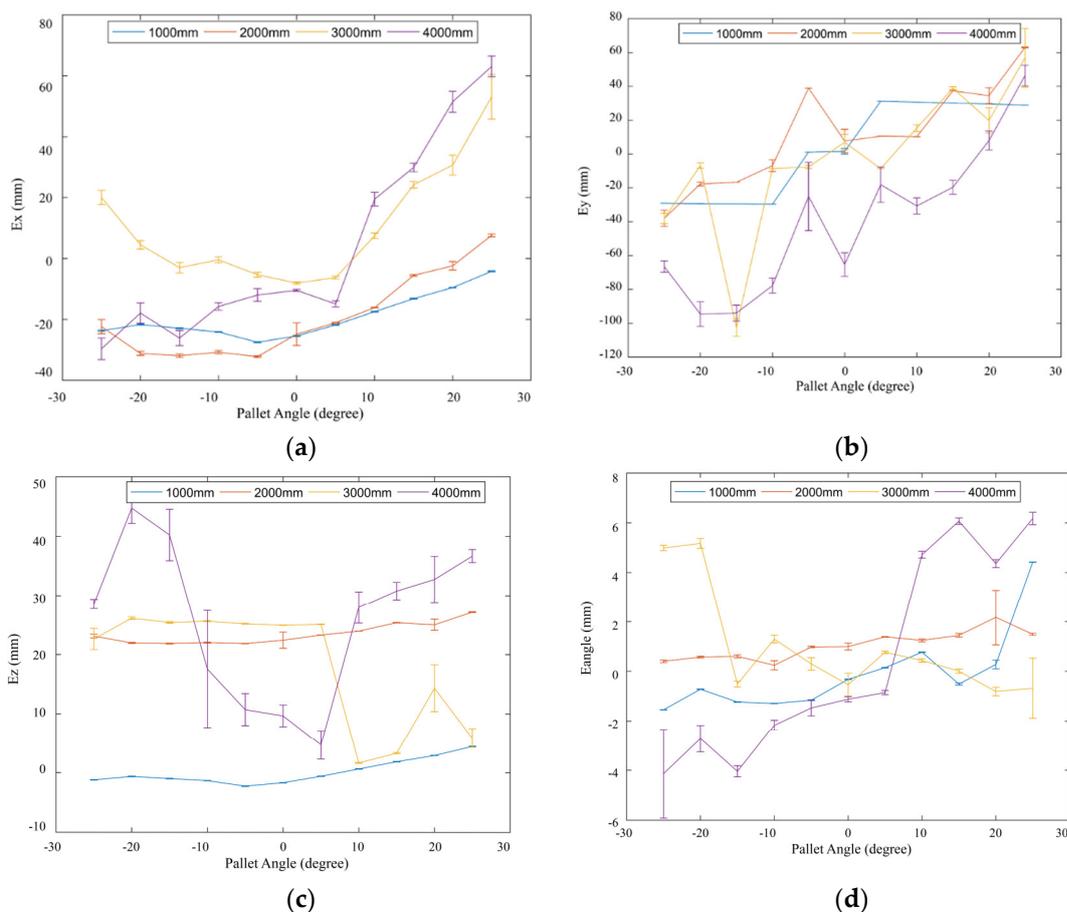
Figure 9. Measurement error curve of the algorithm for different distances and angles of the pallet. (**a**) Curve chart of measurement error in $X-$axis direction versus pallet angle. (**b**) Curve chart of measurement error in $Y-$axis direction versus pallet angle. (**c**) Curve chart of measurement error in $Z-$axis direction versus pallet angle. (**d**) Curve chart of angel measurement error versus pallet angle.

Table 1. Correlation between parameter error and pallet angle.

Distance	COF of Ex	COF of Ey	COF of Ez	COF of Eangle
1000	0.9190	0.8059	0.8000	0.7473
2000	0.8836	0.8320	0.8521	0.8843
3000	0.7580	−0.7414	0.5723	−0.7307
4000	0.8839	−0.0472	0.9140	0.9283

4.4. Accuracy Test in Dynamic Experiment

In Figure 10, the first row shows the coordination of the RGB-D sensor and the detection result of the pallet in {W} space. The third row shows the error of the pallet Ex, Ey, Ez, and Eangle in the approach process.

When the sensor approached the pallet and the angle was in the range 3.08°–7.02°, as shown in Figure 8a, the estimated pallet position was (0.14, −6.74, 58.16) and the angle was 92.01°. The error became smaller when the sensor was closer to the pallet. The maximum errors in the *X* axis, *Y* axis, *Z* axis, and angle were 6.38 mm, 15.92 mm, 54.56 mm, and 2.82°, respectively.

When the sensor approached the pallet along a line, and the angle was in the range 12.41°–19.03°, as shown in Figure 9b, the estimated pallet position was (16.85, −24.12, 61.08), and the angle was 93.39°. The error became smaller when the sensor was closer to the pallet. The maximum errors in the *X* axis, *Y* axis, *Z* axis, and angle were 46.37 mm, 48.42 mm, 33.09 mm, and 5.55°, respectively.

When the sensor approached the pallet along a curve, as shown in Figure 9c, and the angle varied in the range 2.84°–35.98°, the estimated pallet position was (3.87, 40.50, 51.62), and the angle was 90.91°. The error increased when the angle increased. The maximum error in the *X* axis, *Y* axis, *Z* axis, and angle were 47.80 mm, 102.70 mm, 32.74 mm, and 6.51°, respectively.

In the experiments, the angle changed in different ranges. The algorithm showed better performance, and the accuracy was less than 50 mm when the angle was steady and changed in ±20°. When the angle changed in a large range, the *Y* axis error had a dramatic variation because of the drift of pallet center recognition. As shown in Figure 9c, the algorithm recognized the pallet even when the angle reached 35.98°, and the maximum Y error was 102.70 mm. The performance in the dynamic test matched the tendency observed in the static test.
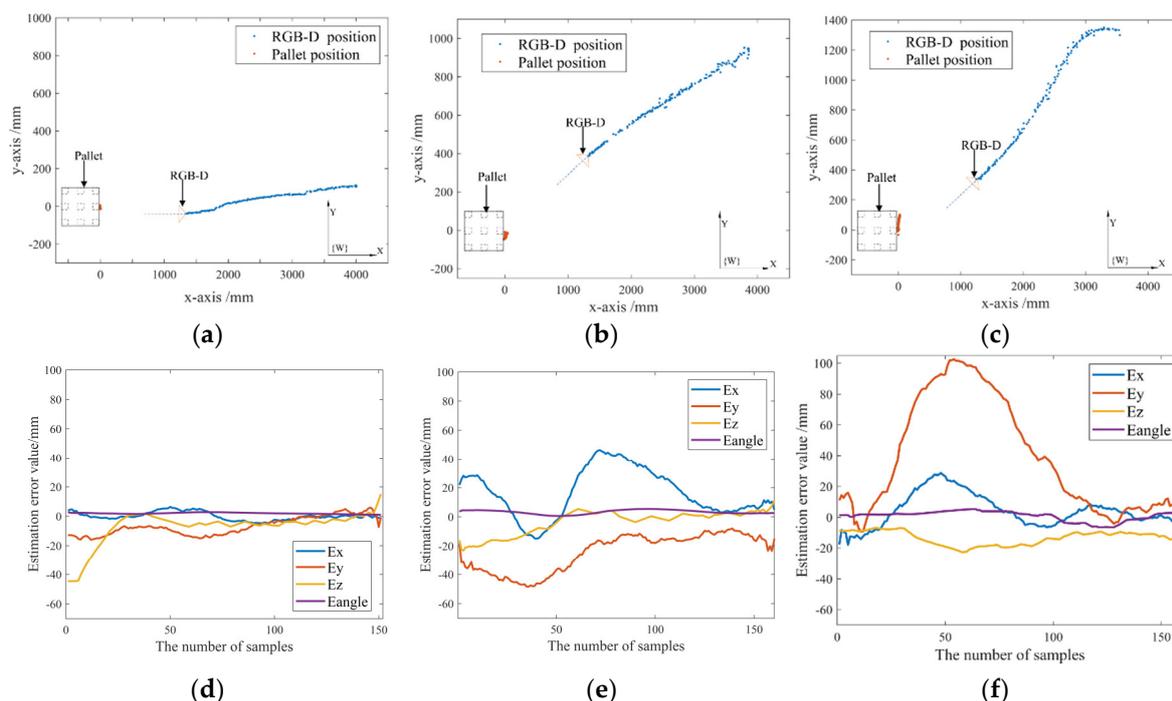
Figure 10. Verification of the algorithm's performance in measuring accuracy when the angle dynamics change. (**a**) Sensor position as well as pallet positioning results data in the sensor angle range of 0−10° during sensor approach. (**b**) Sensor position as well as pallet positioning results data in the sensor angle range of 10−20° during sensor approach. (**c**) Sensor position as well as pallet positioning results data in the sensor angle range of 0−40° during sensor approach. (**d**) Error of pallet positioning results corresponding to Figure 9a. (**e**) Error of pallet positioning results corresponding to Figure 9b. (**f**) Error of pallet positioning results corresponding to Figure 9c.

5. Discussion

5.1. Comparative Performance Analysis

The template matching method provided by OpenCV 3.0 has a strict requirement that the template must be similar to the target in the image; pallet images change with angle changes. In addition, those matching methods are designed for gray or color images, not a label matrix. This method shows worse performance when the pallet angle is larger than ±15°. Occasionally, the matching result is easily affected by environmental interference.

In contrast, the proposed algorithm compresses the category matrix and template to reduce the environment effect, and the matching method is a probability calculation that is more suitable for a label matrix. Consequently, the proposed algorithm was shown to perform better in the face of pallet and environmental changes.

Moreover, the grid compression matching algorithm has many advantages. For instance, the label pallet template has more rich information—including spatial relationships among goods, the ground, pallet feet, and pallet holes—than that of an individual pallet. Meanwhile, it is easy to create a template other than a gray or color template because the labeled template is unchanged under different illumination conditions and only has four labels. The labeled template algorithm, with its strong robustness and implementation, could be applied to many object recognition problems.

5.2. Multiple Pallet Recognition Performance in Warehouse

In a warehouse, goods are placed in a very dense arrangement, making it difficult to segment pallets. Moreover, pallets are often made with plastic, so diversity and notable color features are the most important types of information for this pallet recognition algorithm. However, color information is impacted by illumination conditions, and the classifier cannot precisely categorize pixels. To resolve this issue, the labeled template is used as another classifier to identify the pallet and reduce the interference of incorrect classification.

One reason for misrecognition could be that most of the missed pallets in the test were not completely photographed, with matching scores less than the threshold, as shown in Figure 7a,c. Since the size of the template is determined according to the distance, some pallets will not be recognized when the distance between multiple pallets is large, as shown in Figure 7a. Another reason is that the color spaces of different labels overlapped in dark illumination, thus leading to incorrect classification.

The main approaches to improve the accuracy of yolov5 is to increase the number of pallet samples, and moreover, the number of pallet multi-angle images should be added to the samples to ensure the robustness of the algorithm. The work of sample collection and labeling requires more workload.

5.3. Detection Accuracy Performance

Accuracy is an important indicator for a pallet recognition algorithm. The best algorithms should be useful at long detection distances and large observable angles. A static experiment was performed to test the relationship between accuracy and pallet pose parameters. A dynamic experiment was performed to explore the influence of changing distance and angle on the algorithm.

In the static and dynamic accuracy experiments, the detection accuracy was associated with the detection distance and the angle. The error had a positive correlation with angle of the pallet. As the detection distance was near the edge of the depth sensor's effective range, the depth sensor's detection mistakes affected the detection accuracy.

One reason for the increasing amount of detection errors as the pallet angle increased could have been the length change in pallets in the image with the angle change. Only one shape template was created to match the rotation of pallets. The length of the template was longer than that of the pallets in the image according to the projection principle. This led to the pallet foot measure-points moving, and the measurement error became larger as the rotation angle increased.

In the dynamic accuracy experiment, the detection error fluctuated with the angle change. The performance in the dynamic test matched the tendency shown in the static test. There may have been three major reasons for the error in the dynamic test being larger than that in the static experiment.

The first reason was the size error between the template and pallet in images with the change in distance and angle. The second reason was the error in the transformation matrix between the RGB-D sensor and the location sensor, which caused the error to increase when the distance was longer. The final reason was the communication delay between the RGB-D sensor and location sensor, during which location data with noise affected the detection results.

5.4. Real-Time Performance of Proposed Algorithm

The time consumption of the proposed algorithm was tested in the static experiment. The time consumption was lower when the detection distance decreased. There was no relation between the time consumption and the rotation angle of the pallet. Computational resources are the main cost during template matching, and they are determined by the sizes of the template and category matrix. Their sizes are determined by the compression unit size, which is calculated based on the distance according to Equation (5). To accelerate the proposed algorithm, the matching process should be optimized. The time consumption results indicate that the algorithm has the ability to detect pallets in real time when the detection distance is less than 4 m.

However, the developed algorithm still has some shortcomings. First, illumination changes throughout the entire day, and the training samples hardly cover all illumination conditions. Second, when the pallet color is similar to that of the ground or goods, the color space of each category overlaps and leads to incorrect classification. Third, the compression unit size has a significant impact on pallet recognition performance, and it should be carefully selected by considering the size of the pallet, the camera parameters, and the pallet distance.

6. Conclusions

To recognize and locate pallets in a warehouse environment, a real-time approach based on an improved label template matching algorithm with an RGB-D sensor was proposed. A label template, which contains the goods, pallet, ground color, and pallet information, was created to improve the accuracy of the recognition algorithm. A compression matching algorithm was used to enhance the robustness against incorrect categories and to accelerate the matching process. During algorithm operation, pallet parameters are extracted from three pallet feet and smoothed with a sliding average filter. Experiments were conducted to test the performance of the algorithm. The proposed pallet algorithm was shown to have the ability to recognize and segment multiple pallets in a warehouse environment with an accuracy in which the maximum distance and angle estimation error were −101.1 mm and 6.07° in a range from 1 m to 4 m and an angle of within ±25°. The error was found to decrease with smaller distances and angle changes, and time consumption was found to decrease when the detection distance was reduced, ultimately achieving the real-time requirement. The proposed algorithm could provide a reference for designing autonomous engaging systems.

References

1. Zajac, P.; Rozic, T. Energy consumption of forklift versus standards effects of their use and expectations. *Energy* **2022**, *239*, 122187.
2. World Forklift Market 2013–2018 & 2019–2023: Analysis by Class Type, Fuel Type, Application Type, Company and Region. Available online: https://www.globenewswire.com/news-release/2019/03/07/1749546/0/en/World-Forklift-Market-2013-2018-2019-2023-Analysis-by-Class-Type-Fuel-Type-Application-Type-Company-and Region.html (accessed on 10 December 2020).
3. Horberry, T.; Larssona, T.J.; Johnston, I.; Lambert, J. Forklift safety, traffic engineering and intelligent transport systems: A case study. *Appl. Ergon.* **2004**, *35*, 575–581.
4. Forklift Trucks Market—Growth, Trends, Covid-19 Impact, and Forecast (2022–2027). Available online: https://www.mordorintelligence.com/industry-reports/forklift-trucks-market (accessed on 12 August 2021).
5. Stout-Wiegand, N. Characteristics of work-related injuries involving forklift trucks. *J. Saf. Res.* **1987**,*18*, 179-190.
6. Government of South Australia Safe Work SA—High Risk Work. *A Guide to Forklift Safety*; Government of South Australia: 2015. Available online: https://www.safework.sa.gov.au/sites/default/files/forkliftsafety.pdf?v=1527223033.
7. Marsh, S.M.; Fosbroke, D.E. Trends of occupational fatalities involving machines, United States, 1992–2010. *Am. J. Ind. Med*. **2015**, *58*, 1160–1173.
8. Miller, B.C. Forklift safety by design. *Prof. Saf.* **1988**, *33*, 18.
9. Sarupuri, B.; Lee, G.A.; Billinghurst, M. An augmented reality Guide for assisting forklift operation. In Proceedings of the 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), Merida, Mexico, 19–23 September 2016; pp. 59–60.
10. Occupational Safety and Health Administration (OSHA). OSHA Fatality and Catastrophe Investigation Summaries. 2018. Available online: https://www.osha.gov/pls/imis/accidentsearch.html? (accessed on 26 August 2018).
11. Choi, M.; Ahn, S.; Seo, J. VR-Based investigation of forklift operator situation awareness for preventing collision accidents. *Accid. Anal. Prev.* **2020**, *136*, 105405.
12. Renhu, X. *Research on Path Planning of Intelligent Vehicle Based on Improved A Algorithm and Artificial Potential Field*; Xidian University: Xi'an, China, 2019.
13. An, Y. *Research on Structure Design and Path Planning of Automatic Electric Forklift Truck*; Xi'an University of Technology: Xi'an, China, 2021.
14. Ren, Z.; Lai, J.; Wu, Z.; Xie, S. Deep neuralnet works-based real-time optimal navigation for an automatic guided vehicle with static and dynamic obstacles. *Neurocomputing* **2021**, *443*, 329–344.
15. Zhang, J.; Xia, Y.; Shen, G. A novel learning-based global path planning algorithm for planetary rovers. *Neurocomputing* **2019**, *361*, 69–76.
16. Jia F, Tao Z, Wang F. Wooden pallet image segmentation based on Otsu and marker watershed[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1976(1): 012005.
17. Digani, V.; Hsieh, M.A.; Sabattini, L.; Secchi, C. Coordination of multiple agvs: A Quadratic optimization method. *Auton. Robot.* **2019**, *43*, 539–555.
18. Mercy, T.; VanParys, R.; Pipeleers, G. Spline-based motion planning for Autonomous guided vehicles in a dynamic environment. *IEEE Trans. Control Syst. Technol.* **2017**, *26*, 2182–2189.
19. Baglivo, L.; Biasi, N.; Biral, F.; Bellomo, N.; Bertolazzi, E.; Lio, M.D.; De Cecco, M. Autonomous Pallet Localization and Picking for Industrial Forklifts: A Robust Range and Look Method. *Meas. Sci. Technol.* **2011**, *22*, 1312–1327.
20. Walter, M.R.; Karaman, S.; Frazzoli, E.; Teller, S. Closed-Loop Pallet Manipulation in Unstructured Environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 5119–5126.
21. Bostelman, R.; Hong, T.; Chang, T. Visualization of Pallets. *Opt. East* **2006**, *6384*, 78–89.
22. Pages, J.; Armangué, X.; Salvi, J.; Freixenet, J.; Martí, J. A Computer Vision System for Autonomous Forklift Vehicles in Industrial Environments. In Proceedings of the 9th Mediterranean Conference on Control and Automation MEDS, Dubrovnik, Croatia, 27 June–29 June 2001; pp. 1–6.
23. Byun, S.; Kim, M. Real-Time Positioning and Orienting of Pallets Based on Monocular Vision. In Proceedings of the IEEE International Conference on TOOLS with Artificial Intelligence, Dayton, OH, USA, 3–5 November 2008; pp. 505–508.
24. Tamba, T.A.; Hong, B.; Hong, K.S. A Path Following Control of an Unmanned Autonomous Forklift. *Int. J. Control Autom. Syst.* **2009**, *7*, 113–122.
25. Cintas, R.; Manso, L.J.; Pinero, L.; Bachiller, P. Robust Behavior and Perception using Hierarchical State Machines: A Pallet Manipulation Experiment. *J. Phys. Agents* **2011**, *5*, 35–44.
26. Chen, G.; Peng, R.; Wang, Z.; Zhao, W. Pallet Recognition and Localization Method for Vision Guided Forklift. In Proceedings of the 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 21–23 September 2012; pp. 1–4.
27. Varga, R.; Nedevschi, S. Vision-Based Autonomous Load Handling for Automated Guided Vehicles. In Proceedings of the 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 4–6 September 2014; pp. 239–244.
28. Seelinger, M.; Yoder, J.D. Automatic Visual Guidance of a Forklift Engaging a Pallet. *Robot. Auton. Syst*. **2006**, *54*, 1026–1038.

29. Lecking, D.; Wulf, O.; Wagner, B. Variable Pallet Pick-Up for Automatic Guided Vehicles in Industrial Environments. In Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, Prague, Czech Republic, 20–22 September 2006; pp. 1169–1174.

30. Aref, M.M.; Ghabcheloo, R.; Kolu, A.; Mattila, J. A Multistage Controller With Smooth Switching for Autonomous Pallet Picking. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 2535–2542.

31. Bellomo, N.; Marcuzzi, E.; Baglivo, L.; Pertile, M.; Bertolazzi, E.; De Cecco, M. Pallet Pose Estimation withLIDAR and Vision for Autonomous Forklifts. *IFAC Proc*. **2009**, *42*, 612–617.

32. Varga, R.; Nedevschi, S. Robust Pallet Detection for Automated Logistics Operations. In Proceedings of the International Conference on Computer Vision Theory and Applications, Rome, Italy, 27–29 February 2016; pp. 470–477.

33. Nygards, J.; Hogstrom, T.; Wernersson, A. Docking to Pallets With Feedback From a Sheet-of-Light Range Camera. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Takamatsu, Japan, 31 October–5 November 2000; Volume 3, pp. 1853–1859.

34. Gao, L.; Yuan, P.;Wang, T.; Shi, Z.; Cao, S.; Ji, X. Automatic Recognition About Pallet Based on Tracking Algorithm of Ladar and Slam. In Proceedings of the International Conference on Advanced Robotics and Mechatronics, Macau, China, 18–20 August 2016; pp. 196–201.

35. Oh, J.Y.; Choi, H.S.; Jung, S.H.; Kim, H.S.; Shin, H.Y. An Experimental Study of Pallet Recognition System Using Kinect Camera. *Mob. Wirel.* **2013**, *42*, 167–170.

36. Lienhart, R.; Maydt, J. An Extended Set of Haar-Like Features for Rapid Object Detection. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002.

37. Syu, J.L.; Li, H.T.; Chiang, J.S.; Hsia, C.H.; Wu, P.H.; Hsieh, C.F.; Li, S.A. A Computer Vision Assisted System for Autonomous Forklift Vehicles in Real Factory Environment. *Multimed. Tools Appl.* **2017**, *76*, 18387–18407.

38. Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Experimental Robotics*; Springer: Berlin/Heidelberg, Germany, 2014.

39. Huang, A.S.; Bachrach, A.; Henry, P.; Krainin, M.; Maturana, D.; Fox, D.; Roy, N. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *Robotics Research*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 235–252.

40. Lai, K.; Bo, L.; Ren, X.; Fox, D. A large-scale hierarchical multi-view rgb-d object dataset. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011.

41. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012.

42. Tsiogas E, Kleitsiotis I, Kostavelis I, et al. Pallet Detection and Docking Strategy for Autonomous pallet truck AGV operation[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021: 3444-3451.

43. Mok C, Baek I, Cho Y S, et al. Pallet Recognition with Multi-Task Learning for Automated Guided Vehicles[J]. Applied Sciences, 2021, 11(24): 11808.

44. Luo, J.; Gwun, O. A Comparison of SIFT, PCA-SIFT and SURF. *Int. J. Image Process.* **2013**, *3*, 143–152.

45. Wang, X.; Han, T.X.; Yan, S. An HOG-LBP human detector with partial occlusion handling. In Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; Volume 30, pp. 32–39.

46. Keerthi, S.S.; Shevade, S.K.; Bhattacharyya, C.; Murthy, K.R.K. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Comput.* **2014**, *13*, 637–649.

47. Wang, S.; Lu, H.; Yang, F.; Yang, M.H. Superpixel tracking. Computer Vision (ICCV). In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1323–1330.

48. Li, J.; Chen, X.; He, Z. Adaptive stochastic resonance method for impact signal detection based on sliding window. *Mech. Syst. Signal Processing* **2013**, *36*, 240–255.

49. Papandreou, G.; Kokkinos, I.; Savalle, P.E. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.