

## Article

# Variational Online Learning Correlation Filter for Visual Tracking

Zhongyang Wang, Feng Liu and Lizhen Deng \*

School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 2020010110@njupt.edu.cn (Z.W.); liuf@njupt.edu.cn (F.L.)

\* Correspondence: denglizhen@njupt.edu.cn

**Abstract:** Recently, discriminative correlation filters (DCF) have been successfully applied for visual tracking. However, traditional DCF trackers tend to separately solve boundary effect and temporal degradation problems in the tracking process. In this paper, a variational online learning correlation filter (VOLCF) is proposed for visual tracking to improve the robustness and accuracy of the tracking process. Unlike previous methods, which use only first-order temporal constraints, this approach leads to overfitting and filter degradation. First, beyond the standard filter training requirement, our proposed VOLCF method introduces a model confidence term, which leverages the temporal information of adjacent frames during filter training. Second, to ensure the consistency of the temporal and spatial characteristics of the video sequence, the model introduces Kullback–Leibler (KL) divergence to obtain the second-order information of the filter. In contrast to traditional target tracking models that rely solely on first-order feature information, this approach facilitates the acquisition of a generalized connection between the previous and current filters. As a result, it incorporates joint-regulated filter updating. Through quantitative and qualitative analyses of the experiment, it proves that the VOLCF model has excellent tracking performance.

**Keywords:** model confidence; Kullback–Leibler divergence; spatial temporal regularization; ADMM

**MSC:** 68U10



**Citation:** Wang, Z.; Liu, F.; Deng, L. Variational Online Learning Correlation Filter for Visual Tracking. *Mathematics* **2024**, *12*, 1818. <https://doi.org/10.3390/math12121818>

Academic Editor: J. Alberto Conejero

Received: 10 April 2024

Revised: 19 May 2024

Accepted: 27 May 2024

Published: 12 June 2024



**Copyright:** © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Visual object tracking is an important aspect of remote sensing that has become widely used [1]. Visual object tracking aims to automatically acquire the states of objects in subsequent video frames based on the initial state (center location and scale) of a target given in a video sequence. Vision-based tracking has attracted much attention in the field of computer vision. Over the past decade, with the rapid advancement of related research, numerous tracking methods have emerged, demonstrating highly effective outcomes [2–4]. Advances in visual tracking have been applied in many areas, such as UAV-based monitoring [5], intelligent surveillance, and airplane tracking [6]. However, improving the robustness and accuracy of target tracking algorithms is challenging due to appearance changes caused by rapid motion, occlusion, and scale changes [7,8].

After years of research and development, target tracking algorithms have been divided into two main categories: generative models and discriminative models. With the development of feature extraction technology, the discriminant method has become the mainstream research direction in the field of target tracking due to its good performance. Discriminative models construct an objective function that can significantly distinguish between the background and the target, and tracking models can obtain the exact position of the target object in each frame of the image in a complex environment [9]. In addition, ensuring the robustness of tracking during a long-term target-tracking process is difficult. Therefore, researchers have introduced the concept of online learning. The tracker learns the updated image features online to distinguish the target object from the background.

To address long-term tracking, Kalal et al. [10] proposed a TLD model, which divides the tracking process into three parts: tracking, learning, and detection. The main advantage of this method is the ability to learn more information and avoid repeated mistakes. However, if the target object is rotated out of the original plane, the TLD cannot provide good results. Yu et al. [11] proposed a collaborative training technology that combines a generative model and a discriminative model. The technology uses the subspace features of online learning to model and learn the appearance of the target object and then uses a support vector machine (SVM) classifier to discriminate the objects. This model is very efficient but cannot address occlusion problems.

With the development of correlation filters, Bolme et al. added correlation filters (CFs) to the tracking process [12] and proposed a tracking model based on the minimum output sum of squared error (MOSSE) filter. Heriques et al. introduced a circular structure kernel (CSK) algorithm [13] that relies on illumination intensity features. Kernelized CFs (KCFs) were developed to use more robust features such as the histogram of oriented gradients (HOG) [14]. The discriminative correlation filter (DCF)-based tracker in [15] mitigated two major problems in the existing paradigm: the spatial boundary effect and temporal filter degeneration. The efficient appearance learning models in DCF have been proven to be effective in visual tracking [16]. The spatially regularized DCF (SRDCF) [17] method learned filters from training examples with rigid spatial constraints. Then, spatial-temporal regularized correlation filters (STRCFs) [18] were introduced that included spatial-temporal regularization to handle boundary effects, and they achieved superior performance to SRDCF. However, the STRCF simply measures passive-aggressive learning from the current filter and previous filter via the Euclidean distance. Learning adaptive discriminative correlation filters (LADCFs) [15] exploit the complementary information of the target and background to adaptively optimize the most discriminative spatial features. This approach enables the continuously updated tracker to always be in a lower-dimensional manifold space by combining with explicit distance-based temporal constraints. However, the limitations of explicit distance-based measurements under two-frame filters are inevitable in amplifying overfitting from limited training samples in visual tracking tasks. Recently, joint spatial-temporal feature information has also been found to strongly affect target tracking performance. Traditional tracking methods [19,20] were organized to avoid boundary effects and temporal degradation problems in a similar manner for visual tracking.

Although basing a tracking model on spatiotemporal features improves the performance of the target tracking model, these methods separately embed only temporal filtering information and spatial weighting to regularize traditional DCF trackers. Generally, the target position in a video sequence will always change over time and cannot be predicted, especially if it encounters motion blur or partial occlusion. Since the features extracted by the tracking model are derived from the current frame, the learned tracker highly depends on the feature quality of offline training, instead of relying on the spatiotemporal joint features of the image. Because the motion information between two video frames is disregarded, temporal and spatial consistency is not achieved. Here, we can impose a prior condition in advance, which is that the higher-order derivatives in the learned temporal feature space are small. Based on this finding, the concept of time-consistent slow feature analysis (SFA) [21] in a video can be employed as free supervision [22] to promote spatial-temporal consistency (that is, subtle feature differences between nearby frame pairs) and to ensure that the spatial representation changes smoothly over time. This approach motivates the concept of modeling consistent spatial-temporal dependency for visual tracking. Inspired by these findings, we introduce a model confidence term in our model to mitigate independent spatial-temporal dependency and integrate the spatial information of the previous frame to supplement the temporal dependency information of the current frame without negatively affecting online learning.

Therefore, we propose an improved tracking model, namely the variational online learning correlation filter for visual tracking (VOLCF). Figure 1 shows the process of our proposed variational online learning update in visual tracking. VOLCF improves the

extraction and online learning of spatiotemporal features of image patches by filtering the general target model. VOLCF uses online learning to obtain the feature matrix of the target image block in the previous frame, that is, the trained convolution kernel. Then, the convolution kernel of the previous target image block is convolved with the candidate modules in the candidate set (which can be regarded as template matching) and compared with the filter response graph to obtain the candidate image block with the highest similarity to the original target image block as the target block of the current frame. On this basis, the model confidence term introduced by the VOLCF model can adjust the filter to a reasonable object range that is suitable for tracking, better dividing the area of the target candidate block, and obtaining a more accurate search range. Therefore, the model confidence term reflects the quality of the filter learned by the model, which can achieve more accurate target positioning and improve the scalability of the model in a deformed environment such as during fast motion. Furthermore, considering that in a complex environment, a model that relies only on first-order information easily fails due to sudden changes in the target object, we try to extract second-order information related to the correlations between spatial features and temporal features. Correlation-related second-order features are used to improve the robustness of tracking in complex situations, which is achieved mainly by Kullback–Leibler (KL) divergence. The KL divergence cannot be understood as the “distance”; it is used to measure the information loss of one distribution compared to another. Specifically, if the information loss between the candidate and the target in the previous frame reaches a minimum value, the corresponding candidate block position is the new target position. By constantly changing the parameters of the estimated distribution, we can obtain different values of KL divergence. If the KL divergence reaches the minimum value, the corresponding parameter is the optimal parameter of interest. For the specific value of the parameter, our model uses the covariance matrix to obtain it. In the GAN network [23], the KL divergence is substituted into the objective function to solve the minimax game problem. Mapping to the tracking problem, we build a loss function based on minimizing the information loss of the KL divergence, and the model can be optimized and updated by solving the minimum value of the loss function [24]. In [25], the KL divergence is minimized to train the regression network.

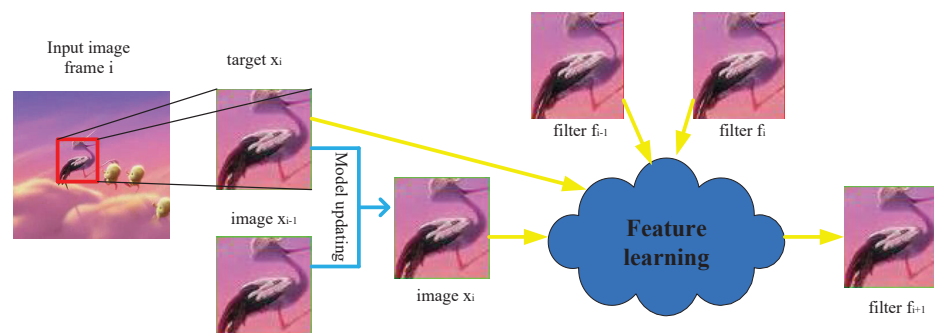


**Figure 1.** Process of online learning update of the filter. All the processing steps involve introducing a confidence term in the new proposed model with previous spatial information  $x_{i-1}$  and performing novel alignment measurements (KL divergence) between the temporal  $f_i$  and  $f_{i-1}$  filters.

Furthermore, we introduce the alternating direction method of multipliers (ADMM) to solve the optimization in the iteration of the subproblems for efficient online learning of our VOLCF. The experimental results obtained on OTB [26], VOT [27] and other benchmarks demonstrate the accuracy and superiority of VOLCF. VOLCF has made considerable progress in terms of robustness and accuracy in comparison with state-of-the-art trackers. Compared with traditional online learning DCF trackers, our VOLCF has several significant advantages in making the VOLCF more interpretable than previous trackers. In summary, we list our main contributions as the following three items.

- VOLCF introduces a model confidence item based on the general model and uses the spatial information of the previous frame to supplement the timing information of the current frame to achieve the precise positioning of the target, thereby ensuring the consistency of the model's spatial-temporal dependency.
- The KL divergence is used to implement the second-order information of the model filter to ensure the consistency of the spatial-temporal information. The parameters are selected and adjusted through the covariance matrix to improve the robustness of the tracking model in complex environments.
- The loss function that was constructed based on the KL minima game is optimized by the ADMM algorithm, which greatly reduces the computational complexity of the model. Our algorithm achieves outstanding performance in both accuracy and robustness against state-of-the-art trackers.

The main structure of the paper is as follows: Section 1 is the introduction of this paper. The related works are introduced in Section 2. We introduce the STRCF tracking formulation in Section 3. Next, the new method proposed in this paper, VOLCF, is deduced and calculated in Section 4. Section 5 analyzes the experiments and results of VOLCF and the state-of-the-art trackers. A brief conclusion of our work is given in Section 6.

## 2. Related Work

### 2.1. Discriminative Correlation Filter-Based Tracking

DCF-based trackers have recently attracted widespread attention. In the frequency domain, the DCF utilizes a circular structure to solve a ridge regression problem. A minimum output sum of squared error (MOSSE) filter [12] was proposed for the first time for visual tracking of grayscale images at a speed of several hundred frames/second, demonstrating the powerful potential of the filter. Moreover, CSK [14] extends dense sampling and kernel tricks based on MOSSE. Staple [28] improved the reliability of the final response by generating a response map with color histograms from the foreground and background. To handle scale variations, the SAMF [29] and DSST [30] algorithms, which perform scale selection in a scale pool after the tracking stage, were proposed. fDSST [31] was also proposed to detect scale during tracking, enhancing effectiveness through combined estimation of scale and location. For the first time, the GFSDCF method [32] suggested selecting channels for DCF-based tracking, and feature selection was conducted across both spatial and channel dimensions. This was performed to assess the structural correlation among the multichannel features of the filter system, consequently enhancing tracking performance.

Moreover, the boundary effects caused by circular shifts significantly impact the tracking performance. BACF based on HOG features was proposed in [33] to acquire fewer boundary effects with correlation filters. For spatial regularization, a predefined filter weighting strategy was proposed to concentrate the filter energy in the central region in SRDCF [17]. The STRCF tracker [18] efficiently solves the boundary effect with spatial-temporal regularization, which is too complex to achieve good tracking performance. LADCF [15] uses the lasso method to adjust and select spatial features. Through adaptive spatial feature selection and temporal consistency constraints, the tracker can perform joint spatial-temporal filtering learning in low-dimensional discriminant manifolds.

In addition, the adaptive decontamination of the training set was proposed in [34] for improving the generalization performance to achieve adaptive multiframe learning in the DCF paradigm. Subgrid tracking was introduced through the study of continuous convolution operators (CCOTs) in [35]. Efficient convolution operators (ECOs) [36] were developed as a lightweight alternative to CCOT, incorporating a generative sample space and dimension reduction mechanism. By jointly modeling the spatial context and historical target information, the STCAT [37] could not only adapt to appearance changes but also maintain a relatively stable filter due to the small variation in targets between frames. In addition, several trackers, such as SiamFC [38], CF-Net [39], HDT [40], and HCF [41], which use deep features, have also been proposed and have shown good tracking performance. Considering the complex changes in the environment, Qi et al. [42] adopted adaptive fusion

methods with deep features to achieve high tracking performance. The JCRCF [43] adds a regularization term to the conventional DCF framework, which can adaptively increase the impact of reliable channels, downweight the corrupted channels and introduce a joint learning strategy to increase the discriminative ability of the filter.

## 2.2. Spatial–Temporal Regularization Tracking Method

The tracking performance is always reduced because of the periodic repetitions on boundary positions. Thus, several spatial regularization methods have been introduced in [44,45] to alleviate unwanted boundary effects. These methods, however, are unable to exploit the circulation structure during learning, resulting in increased computational costs. The recent regularization algorithm in [46,47] increased tracker speed by obtaining useful data from time series relations to multiple channels.

Because of the continuous movement of the target, such a spatial–temporal regularization strategy can play an important role in target tracking. The temporal consistency dictionary learning tracking method proposed in [48] considers temporal consistency and restricts the difference in object appearance in adjacent image frames by adding a temporal regularization term to the object function. In addition, the SRCTRCF tracker proposed by Lei Pu et al. [49] introduces temporal regularization into the objective function to further improve the long-term tracking performance. Moreover, the STCAT [37] jointly models the spatial and temporal information of the target to balance the tradeoff between the appearance variants and long-term memory.

## 2.3. First-Order and Second-Order Online Learning

Online learning was exploited in [18], which artfully embeds temporal regularization to passively make the updated classifier analogous to the previous classifier and aggressively guarantees that the new instance is correctly classified. In addition, online learning has been applied in many large-scale applications, such as learning support vector machines from billions of data [50]. To improve the association between learning and cost-sensitive classification, the effective cost-sensitive online classification framework [51] was recently proposed for optimizing predefined, cost-sensitive metrics. However, traditional online learning methods learn only first-order information, which is insufficient because it suffers from overfitting to unconditional appearance variants. The learning algorithm [52] renders the updated classifier similar to the previous classifier and guarantees that the new instance is correctly classified.

In this paper, we use second-order information for online learning procedures involving visual tracking to enhance the prediction performance of classification models. This approach prevents rigid deformation and severe appearance variants.

## 3. Spatial–Gemporal Regularization Tracking Formulation

A training set of samples  $X = [x_1, x_2, \dots, x_D] \in \mathbb{R}^{M \times N \times D}$  with the Gaussian-shaped regression labels  $y = [y_1, y_2, \dots, y_D]$  is given, and the filter is defined as  $f$  with  $D$  channels. Our goal is to learn a function differentiating the target from the surrounding environment. DCF trackers can utilize the fast Fourier transform (FFT) and its inverse transform ( $F^{-1}$ ) to improve the computational efficiency in the Fourier domain.

$$\psi(X; f) = f^T X = f \otimes x = F^{-1}(\hat{f} \odot \hat{x}^*) \quad (1)$$

where  $\hat{x}$  is the Fourier representation of  $x$  and  $\hat{x}^*$  is the complex conjugate of  $\hat{x}$  in the frequency domain.  $\otimes$  denotes the convolution operator, and  $\odot$  denotes the operator of elementwise multiplication. Upon acquiring the tracked feature of the target, the updated model undergoes training by minimizing the subsequent loss function.

$$\tilde{f} = \arg \min_f \theta(y, H) + \varphi(f) \quad (2)$$

where  $\theta(\cdot)$  is the objective and  $\varphi(\cdot)$  is the regularization term.  $H = (X, y)$  represents the labeled training samples created by the circulant matrix, with the base sample  $x$  centered on the tracking result in the current frame. In the learning phase of traditional DCF target tracking, the ridge regression problem is usually formed with the quadratic loss and  $\ell_2$ -norm penalty. Thus,  $\theta(y, H) = \|f^T X - y\|_2^2$  and  $\varphi(f) = \|f\|_2^2$  are commonly employed.

DCF trackers can identify the optimal candidate to maximize the discriminant function in a given filter based on model parameter  $\tilde{f}$  and prior knowledge:

$$\tilde{x}_i = \arg \max_{x_i} \psi(x_i; \tilde{f}) \quad (3)$$

where the candidate is a feature map extracted from the image.

Because the tracking target is constantly moving, there is a potential relationship in the time series of the target during movement. To highlight the temporal characteristics of the filter, the subscript  $t - 1$  is used to represent the previous filter. Target tracking may benefit from a spatial-temporal regularization strategy [18] due to its continuous movement. The objective function of the improved spatial-temporal DCFs is expressed in the following general form:

$$f = \arg \min_f \frac{1}{2} \left\| \sum_{d=1}^D x_t^d * f^d - y \right\|_2^2 + \frac{1}{2} \sum_{d=1}^D \left\| \omega \cdot f^d \right\|_2^2 + \frac{\mu}{2} \|f - f_{t-1}\|_2^2 \quad (4)$$

where  $x$  denotes input samples,  $D$  is the number of channels;  $f \in \mathbb{R}^{M \times N \times D}$  denotes the current frame filter,  $M, N$  is the size of the filter,  $f_{t-1}$  denotes the previous frame filter,  $y$  is the Gaussian-shaped labels,  $\omega$  is the spatial regularization weight matrix, and  $\mu$  is the time regularization parameter.

#### 4. Proposed VOLCF Method

The STRCF model is shown in Equation (4). Our model improves the time regularization term of STRCF and uses prior information to supplement the timing information of the present frame to achieve a precise target location. The specific modeling process of this article is as follows:

- Firstly, online learning is used to obtain the feature matrix of the target image block, that is, training the convolutional kernel. Then, the convolutional kernel of the previous target image block is convolved with the candidate modules in the candidate set, and the candidate image block most similar to the original target image block is obtained as the target block for the current frame by comparing with the filter response map.
- Subsequently, the model confidence term introduced by the VOLCF model can adjust the filter to a reasonable object range to adapt to track, better delineate the region of target candidate blocks, and obtain a more accurate search range.
- Additionally, in order to improve the robustness of tracking in complex situations, the article also attempts to extract second-order information related to spatial and temporal feature correlations, implemented through KL divergence, which is used to measure the information loss between one distribution and another.
- Finally, the article adopts the alternating direction method of multipliers (ADMM) to solve the iterative optimization of subproblems, thus achieving efficient online learning of VOLCF.

##### 4.1. Framework of the VOLCF Model

We assume that the current filter  $f_t$  and previous filter  $f_{t-1}$  obey the Gaussian distribution ( $N$ ) with a mean filter and covariance matrix  $\Phi$ . Thus, we improve the loss functions by minimizing the following unconstrained objective:

$$f = \arg \min_f \frac{1}{2} \left\| \sum_{d=1}^D x_t^d * f^d - y \right\|_2^2 + \frac{1}{2} \sum_{d=1}^D \left\| \omega \cdot f^d \right\|_2^2 + R_{MC} + R_{KL}(N(f, \Phi) \| N(f_{t-1}, \Phi_{t-1})) \quad (5)$$

where  $R_{KL}$  is the Kullback–Leibler (KL) divergence and the covariance weight matrix  $\Phi$  based on KL divergence can be understood as a confidence-weighted matrix.  $R_{MC}$  is the model confidence term, which is introduced in the proposed model to estimate the covariance weight matrix, improving the target tracking appearance and success rate. Thus, we utilize second-order, covariance-weighted online learning to maintain a probabilistic measure of confidence in each channel of the filter.

#### 4.2. Kullback–Leibler Divergence Based on the $f_{t-1}$ Filter

To solve this optimization problem, we need to explicitly describe the Kullback–Leibler divergence. For convenience, we assume that the online model has a multivariate Gaussian distribution, i.e.,  $f \sim N(f_0, \Phi)$ , where  $f_0$  is the mean value vector of the filter distribution and  $\Phi$  represents the covariance matrix of the distribution. In the previous section, the function  $\psi(x_i; f) = f^T \cdot x_i$  indicates the prediction of the class label of sample  $x_i$  in the  $i$ th dimension with the given multivariate Gaussian distribution. For clarity, we regard each value  $f_i$  in the  $i$ -th dimension as the model knowledge about the corresponding feature  $x_i$  and regard the diagonal entry of covariance matrix  $\Phi_{i,i}$  as the confidence of feature  $f_i$ . The smaller the value of  $\Phi_{i,i}$  is, the more confident the learner is in the mean weight value  $f_i$ . In addition to the diagonal values, other covariance terms  $\Phi_{i,j}$  can be taken as the correlations between two weight values  $f_i$  and  $f_j$  for image features  $x_i$  and  $x_j$ . In this way, we model the parameter confidence for each element along all channels with a diagonal Gaussian distribution. Thus, we construct the Kullback–Leibler divergence term based on the divergence between the empirical distribution and the probability distribution. As a result, we obtain  $R_{KL}$  as follows:

$$R_{KL}(N(f, \Phi) \| N(f_{t-1}, \Phi_{t-1})) = \frac{\mu}{2} \sum_{d=1}^D \left[ \log \frac{\det \Phi_{t-1}^d}{\det \Phi^d} + \text{Tr} \left( \left( \Phi_{t-1}^{-1} \right)^d \Phi^d \right) + \left( f_{t-1}^d - f^d \right)^T \left( \Phi_{t-1}^{-1} \right)^d \left( f_{t-1}^d - f^d \right) \right] \quad (6)$$

where  $\mu$  is the penalty factor.  $\Phi^d, \Phi_{t-1}^d$  in  $\mathbb{R}^{MN \times MN}$  is the covariance matrix for the distribution with diagonal  $\Phi$ .  $\det(\cdot)$  and  $\text{Tr}$  values are defined as the matrix determinant and matrix trace, respectively.

#### 4.3. Model Confidence Based on $x_{t-1}$ Frame

In the traditional DCF tracker, the temporal regularization term generally includes only the second term in Equation (6), that is, the Euclidean distance, to describe the filter distribution similarity. This method only uses first-order information in online learning, which generally leads to suboptimal exploitation of available information and a slow convergence rate. Additionally, the model assumes that the class label is always revealed to the learner at the end of each iteration, which is not realistic for real-world applications. The model's confidence is heavily reliant on filter information, which may not be precise in early rounds. We improve our loss function by introducing the model confidence based on  $x_{t-1}$  to achieve high learning accuracy with low labeling loss.

$$\begin{aligned}
 f = \arg \min_f & \underbrace{\frac{1}{2} \left\| \sum_{d=1}^D x_t^d * f^d - y \right\|_2^2}_{\text{data-fidelity-term}} + \underbrace{\frac{1}{2} \sum_{d=1}^D \left\| \omega \cdot f^d \right\|^2}_{\text{space-regularization-term}} \\
 & + \underbrace{\frac{\beta}{2} \sum_{d=1}^D \left( x_{t-1}^d \right)^T \Phi^d x_{t-1}^d}_{\text{model-confidence-term}} + \underbrace{R_{KL}}_{\text{KL-divergence-term}}
 \end{aligned} \quad (7)$$

where  $\beta$  indicates the penalty factor and the covariance matrix  $\Phi$  enhances the similarity between the current filter  $x$  and the previous filter  $x_{t-1}$  to promote the sparsity of the multichannel filter  $f$ ,  $\sum_{d=1}^D \left\| \omega \cdot f^d \right\|^2$  denotes the spatial regularizer. In general, the first term of the formula is data fidelity, the second term is the space penalty, and the third term and fourth term are our improved points, namely, the model confidence term and KL divergence term, respectively. The  $\sum_{d=1}^D \left( x_{t-1}^d \right)^T \Phi^d x_{t-1}^d$  is the confidence value of the model. When the value is small, the classifier has good training on the instances that are similar to  $x_t$ .

#### 4.4. Optimization of VOLCF with ADMM

We optimize Equation (7) by introducing an auxiliary variable, denoted as  $g$ , and a Lagrange multiplier, denoted as  $s$ . We introduce the auxiliary variable  $g$  to ensure equality  $f^d = g^d$ . This process leads us to derive the Lagrangian augmentation function.

$$\begin{aligned}
 L(f, g, s, \Phi) = & \frac{1}{2} \left\| \sum_{d=1}^D x_t^d * f^d - y \right\|^2 + \frac{1}{2} \sum_{d=1}^D \left\| \omega \cdot g^d \right\|^2 \\
 & + \sum_{d=1}^D (f^d - g^d)^T s^d + \frac{\gamma}{2} \sum_{d=1}^D \left\| f^d - g^d \right\|^2 \\
 & + \frac{\beta}{2} \sum_{d=1}^D (x_{t-1}^d)^T \Phi^d x_{t-1}^d + R_{KL}
 \end{aligned} \quad (8)$$

Let  $h = \frac{1}{\gamma}s$ , where  $\gamma$  is a step-size parameter; the above formulation is converted to

$$\begin{aligned}
 L(f, g, h, \Phi) = & \arg \min_{f, \Sigma} \frac{1}{2} \left\| \sum_{d=1}^D x_t^d * f^d - y \right\|^2 + \frac{1}{2} \sum_{d=1}^D \left\| w \cdot g^d \right\|^2 \\
 & + \frac{\beta}{2} \sum_{d=1}^D (x_{t-1}^d)^T \Phi^d x_{t-1}^d + \frac{\gamma}{2} \sum_{d=1}^D \left\| f^d - g^d + h^d \right\|^2 + R_{KL}
 \end{aligned} \quad (9)$$

where  $g^d, h^d$  in  $\mathbb{R}^{M \times N}$  are the same size as  $x^d$ . We use the ADMM to solve the three subproblems as follows:

$$\left\{ \begin{aligned}
 f^{i+1}, \Phi^{i+1} = & \arg \min_{f, \Phi} \left\| \sum_{d=1}^D x_t^d * f^d - y \right\|^2 + R_{KL} \\
 & + \gamma \sum_{d=1}^D \left\| f^d - g^d + h^d \right\|^2 + \beta \sum_{d=1}^D (x_{t-1}^d)^T \Phi^d x_{t-1}^d \\
 g^{i+1} = & \arg \min_{g^d} \sum_{d=1}^D \left\| w \cdot g^d \right\|^2 + \gamma \sum_{d=1}^D \left\| f^d - g^d + h^d \right\|^2 \\
 h^{i+1} = & h^i + f^{i+1} - g^{i+1}
 \end{aligned} \right. \quad (10)$$

#### 4.4.1. Updating $f$

$\Phi, \Phi_{t-1}$  are held constant for updating  $f^{i+1}$ .  $(x_{t-1}^d)^T \Phi^d x_{t-1}^d$  and  $(f_{t-1}^d - f^d)^T (\Phi_{t-1}^{-1})^d (f_{t-1}^d - f^d)$  in  $R_{KL}$  can be rewritten as  $\|x_{t-1}^d\|^2, \|f_{t-1}^d - f^d\|^2$  with weighting coefficients  $\Phi^d$  and  $\Phi_{t-1}^d$ , respectively. Thus, the variable  $f$  is expressed in the frequency domain as

$$\begin{aligned} L(\hat{f}, \Phi) = \arg \min_{\hat{f}, \Sigma} & \left\| \sum_{d=1}^D \hat{x}_t^d \cdot \hat{f}^d - \hat{y} \right\|^2 \\ & + \gamma \sum_{d=1}^D \left\| \hat{f}^d - \hat{g}^d + \hat{h}^d \right\|^2 + \beta \sum_{d=1}^D (\hat{x}_{t-1}^d)^T \Phi^d \hat{x}_{t-1}^d \\ & + \mu \sum_{d=1}^D \left[ \log \frac{\det \Phi_{t-1}^d}{\det \Phi^d} + \text{Tr}((\Phi_{t-1}^{-1})^d \Phi^d) \right. \\ & \left. + (\hat{f}_{t-1}^d - \hat{f}^d)^T (\Phi_{t-1}^{-1})^d (\hat{f}_{t-1}^d - \hat{f}^d) \right] \end{aligned} \quad (11)$$

where  $\hat{f}$  denotes the discrete Fourier transform (DFT) of the filter  $f$ . Equation (11) is decomposed into  $M \times N$  subproblems (the  $j$ -th subproblem is about the  $j$ -th element of  $f$  along with  $D$  channels), and each of them is defined as follows:

$$\begin{aligned} L(\hat{f}, \Phi) = \arg \min_{\hat{f}, \Phi} & \left\| \hat{x}_t^T \hat{f} - \hat{y} \right\|^2 + \gamma \left\| \hat{f} - \hat{g} + \hat{h} \right\|^2 \\ & + \beta \hat{x}_{t-1}^T \Phi \hat{x}_{t-1} + \mu \left[ \log \frac{\det \Phi_{t-1}}{\det \Phi} \right. \\ & \left. + \text{Tr}(\Phi_{t-1}^{-1} \Phi) + (\hat{f}_{t-1} - \hat{f})^T \Phi_{t-1}^{-1} (\hat{f}_{t-1} - \hat{f}) \right] \end{aligned} \quad (12)$$

Taking the derivative of Equation (12) as zero, we obtain the closed-form solution for  $\hat{f}$  and  $\Phi$ :

$$\hat{f} = \frac{\hat{x}_t \hat{y} + \gamma(\hat{g} - \hat{h}) + \mu \Phi_{t-1}^{-1} \hat{f}_{t-1}}{\hat{x}_t^T \hat{x}_t + \gamma + \mu \Phi_{t-1}^{-1}} \quad (13)$$

Equation (13) can be solved with the Sherman–Morrison formula; then, we have

$$\begin{aligned} \hat{f} &= (A^{-1} - \frac{A^{-1} \hat{x}_t \hat{x}_t^T A^{-1}}{1 + \hat{x}_t^T A^{-1} \hat{x}_t}) [\hat{x}_t \hat{y} + \gamma(\hat{g} - \hat{h}) + \mu \Phi_{t-1}^{-1} \hat{f}_{t-1}] \\ A &= \gamma + \mu \Phi_{t-1}^{-1} \end{aligned} \quad (14)$$

#### 4.4.2. Updating $\Phi$

According to the ADMM algorithm,  $\Phi$  in the solution of  $\hat{f}$  obtained by Equation (14) is  $\Phi^{-1}$ ; thus, we need to calculate only the result  $\Phi^{-1}$ . We obtain the derivative of  $\Phi$  from Equation (12).

$$\frac{\partial L}{\partial \Phi} = \mu (-\Phi^{-1} + \Phi_{t-1}^{-1}) + \beta \hat{x}_{t-1} \hat{x}_{t-1}^T = 0 \quad (15)$$

Updating for  $\Phi^{-1}$ , we obtain

$$\Phi^{-1} = \Phi_{t-1}^{-1} - \frac{\frac{\beta}{\mu} \Phi_{t-1}^{-1} \hat{x}_{t-1} \hat{x}_{t-1}^T \Phi_{t-1}^{-1}}{1 + \frac{\beta}{\mu} \hat{x}_{t-1}^T \Phi_{t-1}^{-1} \hat{x}_{t-1}} \quad (16)$$

In the actual calculation process,  $\Phi^{-1}$  is directly substituted into the calculation; thus, we allow the solution for  $\Phi^{-1}$  to implicitly produce a full matrix and then project it to a diagonal matrix, where the nonzero off-diagonal entries are dropped.

#### 4.4.3. Updating $g$

Similarly, the solution of  $g$  is divided into  $M \times N$  subproblems, and the  $j$ -th subproblem concerns the  $j$ -th element of  $f$  along all  $D$  channels. Then, each of them is defined as

$$\hat{g} = \arg \min_{\hat{g}} \left\| \omega^T \hat{g} \right\|^2 + \gamma \left\| \hat{f} - \hat{g} + \hat{h} \right\|^2 \quad (17)$$

Taking the derivative of Equation (17) as zero, we obtain the closed-form solution for

$$\hat{g} = \frac{\gamma}{\gamma - \omega} (\hat{f} + \hat{h}) \quad (18)$$

#### 4.4.4. Updating Online Model

Online model updating plays a very important role in visual object tracking. The object's appearance often changes during the tracking process because of the influence of posture, scale, occlusion, and rotation. Therefore, we utilize an online adaptation update strategy to improve the robustness of our method. The online adaptation model at frame  $t$  is formulated as follows:

$$\hat{x}_t = (1 - \eta) \hat{x}_{t-1} + \eta \hat{x}_t \quad (19)$$

where  $\eta$  is the learning rate, which is used to update the model in the  $t$  frame.

#### 4.5. Computational Complexity and Convergence Analysis

From the previous derivation, Equation (11) is decomposed into  $M \times N$  subproblems, and each is a system of linear equations with  $D$  channels. Thus, each system can be solved in  $\mathcal{O}(D)$ . The complexity of solving  $\hat{f}$  is  $\mathcal{O}(DMN \log(MN))$  when considering the DFT and inverse DFT. The computational cost for  $g$  is  $\mathcal{O}(DMN)$ , which is the same as that for  $\Phi$  and  $x$ . Hence, the overall cost of our algorithm is  $\mathcal{O}(DMN \log(MN) + 3DMN)N_I$ , where  $N_I$  represents the maximum number of iterations.

Based on the previous analysis and derivation, the optimization process is implemented using the ADMM algorithm. Thus, the solution for each optimization subproblem is closed. Therefore, convergence to global optimality is guaranteed, which satisfies the Eckstein–Bertsekas condition [53]. In addition, it is known from experience that the ADMM algorithm can converge in 2 iterations for most sequences, and we set the efficiency of  $N_I$  to 2. The detailed procedure is given in Algorithm 1.

---

#### Algorithm 1 VOLCF tracking algorithm

---

- 1: Step 1. *Input*:  
the initial target bounding box  $b_1$ , the target size  $s_z$ , the search window  $s_{zw}$ , the initial tracking frame  $I_1$ , the model learning rate  $\eta$ , the penalty factor  $\mu$  and  $\beta$ , the step-size parameter  $\gamma$ , filter model  $f$ ;
  - 2: Step 2. *Tracking*:  
t = 1;  
Extract the target features from  $I_1$  with area  $b_1$ ;  
Obtain corresponding feature representations  $x$ ;
  - 3: Step 3. *Learning*:  
t = t + 1;  
Crop the search window with  $s_{zw}$  from the current frame  $I_t$  and extract the features from the search window;  
Obtain multi-channel feature representations  $x$  based on current target bounding box  $b_t$ ;  
Get the target position of the current frame t;
  - 4: Step 4. *Updating*:  
Update filter model  $\hat{f}$  by using Equation (14);  
Update confidence coefficient matrix  $\Phi_t$  by using Equation (16);  
Update feature representations  $\hat{x}_t$  by using Equation (19);
  - 5: Step 5. *Output*:  
The position and scale of the object target in each frame;  
Updated filter model  $\hat{f}$  for frame t.
-

## 5. Experiments and Results

This section describes experiments that were carried out based on several kinds of benchmark datasets to evaluate the performance of VOLCF for visual tracking, including the OTB dataset, TempleColour dataset, VOT dataset, and DTB dataset. Our VOLCF algorithm was implemented in MATLAB 2017a, and all the experiments were run on a PC equipped with an Intel i7 7700 CPU, 32 GB of RAM and a single NVIDIA GTX 1070 GPU. To ensure the authenticity of the experimental data, we used the public code or results provided by the author for a fair comparison.

### 5.1. Experimental Setting and Evaluation Criterion

Following the settings in [18], we set the side length of the square region centered at the target to  $\sqrt{5WH}$  ( $W$  and  $H$  represent the width and height, respectively, of the target scale). The corresponding image area is proportional to the area of the target bounding box. Then, we extracted HOG and deep features from the image region using a cell size of  $4 \times 4$  pixels. To reduce the boundary discontinuities, the features were further weighted by a cosine window. For the ADMM algorithm, we set the hyperparameters in Equation (7) to  $\mu = 16$  throughout all the experiments. The initial step size parameter  $\gamma^{(0)}$ , maximum value  $\gamma^{(\max)}$ , and learning rate  $\eta$  are set to 10, 100, and 1.2, respectively.

In assessing the effectiveness of our VOLCF algorithm, we adopted the one-pass evaluation (OPE) metric, as recommended by the OTB benchmark. Precision plots depict the accuracy rates of predicted positions against ground truth across various thresholds. Success plots gauge performance based on average overlap, considering both the size and position [54].

### 5.2. Analysis of the OTB Dataset

In this section, we compare our VOLCF tracker against 25 state-of-the-art trackers, including HOG features (i.e., STRCF (HOG) and STRCF (HOGCN) [18], ECO-HC [36], SRDCFdecon [34], BACF [33], Staple+CA [55], SRDCF [17], Staple [28], SAMF+AT [56], SAMF [29], MEEM [57], DSST [30], LADCF [15], GFSDCF (HOGCN) [32] and KCF [14]), and deep features (i.e., MDNet, ECO [36], DeepSTRCF [18], HDT [40], HCF [41], Deep-SRDCF [31], SiamFC [38], and CF-Net [39]) based on the Object Tracking Benchmark (OTB).

The OTB dataset in this section includes OTB50, CVPR2013, OTB100, and another general testing dataset, Temple-Colour. The OTB experiments focus on single-object tracking and use two metrics, precision plots and success plots, as mentioned above. The robustness of the experimental results on OTB is judged by 11 attributes, including illumination variation, occlusion, motion blur, in-plane rotation, out-of-view, low resolution, scale variation, deformation, fast motion, out-of-plane rotation and background clutter.

#### 5.2.1. OTB50 Dataset

We compare our method with other state-of-the-art methods on the basis of HOG and deep features on the OTB50 dataset, which contains 50 video sequences. Figure 2 shows the precision and success plots of our VOLCF tracker and 15 state-of-the-art trackers, namely, STRCF (HOG) and STRCF (HOGCN) [18], ECO-HC [36], SRDCFdecon [34], BACF [33], Staple+CA [55], SRDCF [17], Staple [28], SAMF+AT [56], SAMF [29], MEEM [57], DSST [30], LADCF [15], GFSDCF (HOGCN) [32] and KCF [14], with HOG features. Our proposed VOLCF tracker achieves a precision of 0.817 and a success score of 0.607, which is the best performance among all the trackers. Compared with STRCF, which has a precision of 0.811 and a success score of 0.600, our VOLCF tracker has improved by almost 0.74% and 1.17%, respectively.

The success plots for four attributes, including deformation, illumination variation, low resolution, and out-of-plane rotation, are shown in Figure 3. Our VOLCF tracker has the best performance for three attributes, with the exception of the deformation attribute, and achieves improvements of 2.7%, 0.35%, and 4.2% compared with the second tracker. The other trackers do not perform well in such scenes due to the oversight of the potential link of the moving object target in the time series. In addition, for the deformation attribute,

VOLCF outperforms STRCF by 5.1% and has a score that is lower than that of the best tracker by only 0.017%.

Next, we performed the experiments on VOLCF and other trackers with deep features. As shown in Figure 4, VOLCF performs significantly better than most of the competing trackers, with the exception of ECO-HC [36] and MDNet and surpasses its counterpart DeepSTRCF [18] by 3.6% in successful plots of OPE. Compared with Figure 2, the performance of VOLCF with deep features is much better than that with HOG features in both parts of the OPE.

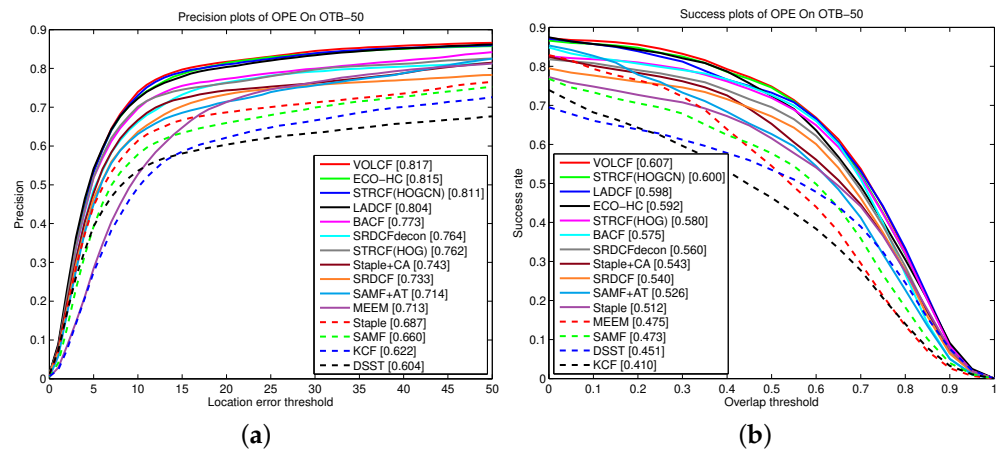


**Figure 2.** Precision and success plots of OPE on the OTB50 dataset with the HOG feature. (a) Precision plots of OPE on OTB50. (b) Success plots of OPE on OTB50.

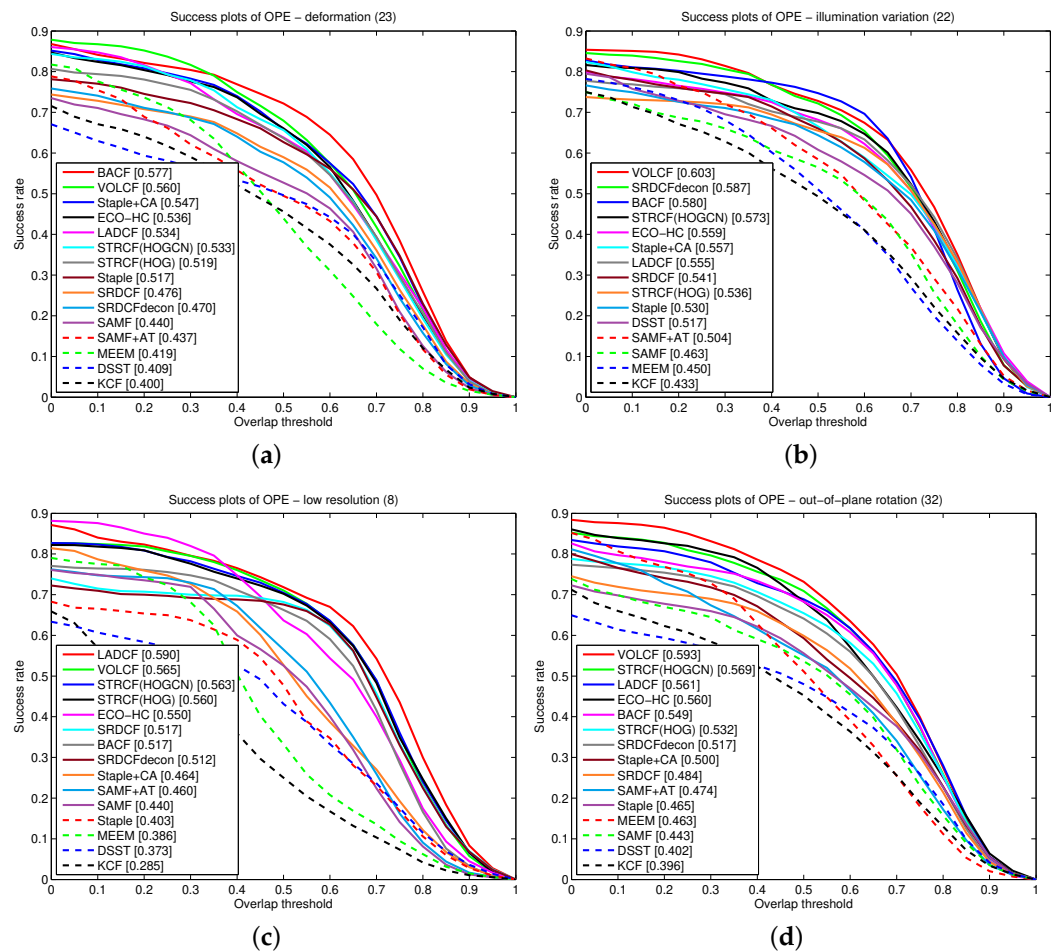


**Figure 3.** Comparison of success plots on OTB50 for the subset of challenging attributes: (a) deformation, (b) illumination variation, (c) low resolution, (d) out-of-plane rotation.

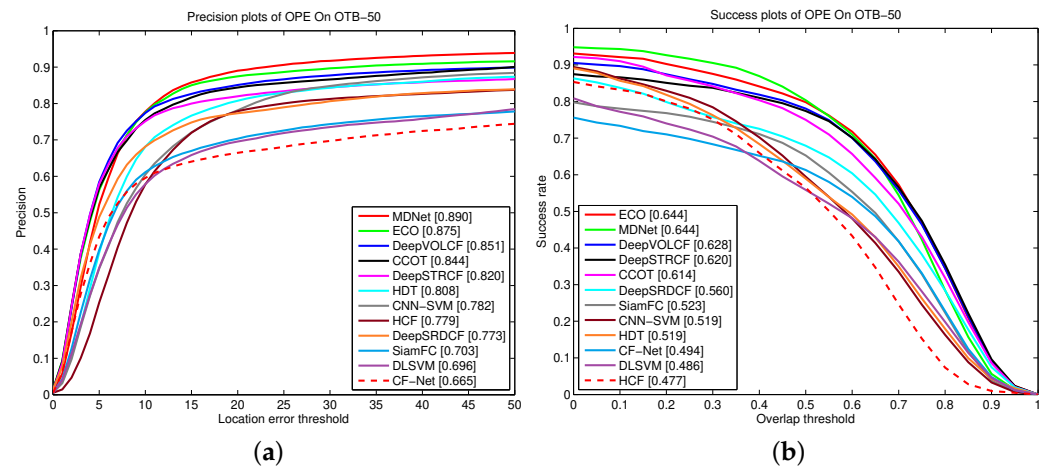


**Figure 4.** Precision and success plots of OPE on the OTB50 dataset with deep features. (a) Precision plots of OPE on OTB50. (b) Success plots of OPE on OTB50.

### 5.2.2. CVPR2013 Dataset

For more thorough evaluation, we evaluated our VOLCF tracker on the CVPR2013 dataset, which includes one more video than OTB50 does, in comparison with 15 state-of-the-art trackers, including ECO [36], CCOT [35], ECO-HC [36], DeepSTRCF, STRCF (HOG), STRCF (HOGCN) [18], LADCF [15], GFSDCF (HOGCN) [32] and DSST [30]. In Figure 5, we note that the VOLCF has the best performance among the trackers with the HOG feature, achieving a score of 0.676, which is 0.45% higher than that of the STRCF and has a high score on the success plot with the deep feature, which is 1.2% higher than that of DeepSTDDCF. Compared with the results in Figure 2, VOLCF outperforms the other methods on CVPR2013 on both HOG and deep features. The scores on CVPR2013 are 0.022 and 0.005 greater than those on OTB50 with HOG and deep features, respectively.

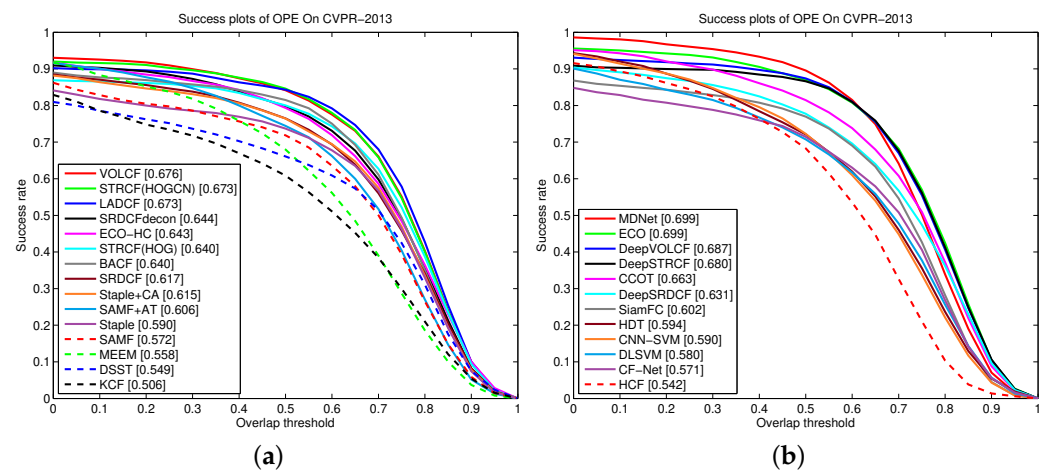


**Figure 5.** Comparison of the overlap success plots with the state-of-the-art trackers on CVPR2013: (a) trackers with HOG features and (b) trackers with deep features.

### 5.2.3. OTB100 Dataset

The OTB100 benchmark consists of 100 fully annotated video sequences. We compare our VOLCF model with the 13 trackers mentioned in OTB50. Figure 6 compares the overlap success plots of four attributes, namely, the deformation, illumination variation, out-of-plane rotation, and scale variation, with the HOG feature. Benefiting from the model confidence and the KL divergence term, our VOLCF algorithm can update the filters with the latest samples in close proximity to both of the previously learned filters. Therefore, the VOLCF tracker has the best performance for these attributes. For example,

our VOLCF method achieves absolute gains of 1.6% over the second-best method in terms of the attributes of out-of-plane rotation.

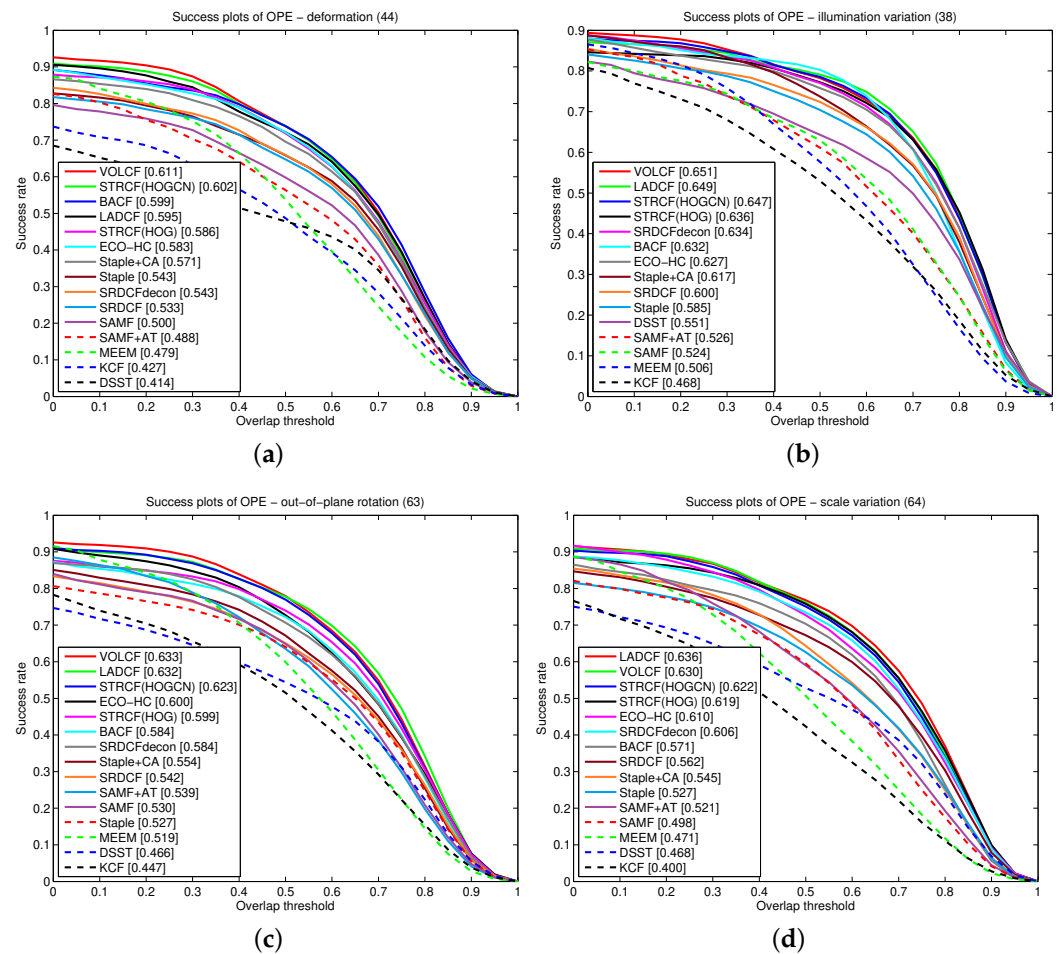


**Figure 6.** Overlap success plots of the competing trackers on the OTB100 dataset for the challenging attributes: (a) deformation, (b) illumination variation, (c) out-of-plane rotation, and (d) scale variance.

Figure 7 shows a comparison of the overlap success plots of our VOLCF tracker with those of other trackers. As shown in Figure 7a, our VOLCF achieves the second-best performance, with a success plot score of 65.4%, outperforming its counterparts STRCF(HOGCN) [18] and STRCF(HOG) by gains of 0.3% and 2.3%, respectively. In the second picture, the VOLCF algorithm achieves the second-best performance among all the trackers, with a score of 68.1%, which is 0.7% lower than that of the ECO [36]. The score of VOLCF with deep features is much greater than the score with HOG features. We obtain the overlap success curves of these trackers with HOG features and deep features, which are ranked using the area under the curve (AUC) score.

As shown in Figures 6 and 7, the superiority of the experimental results based on the OTB100 benchmark is more significant than that based on the OTB50 benchmark. The main reason is that the number of videos based on OTB100 is greater, which is beneficial for target selection. For clarity, the results of VOLCF and five state-of-the-art trackers, namely, STRCF [18], ECO-HC [36], SRDCF [17], SRDCFdecon [34] and BACF [33], are shown in Figure 8, which contains the tracking results of six video sequences. As shown in Figure 8, VOLCF successfully captures the target, and the tracking accuracy is higher than that of the other trackers.

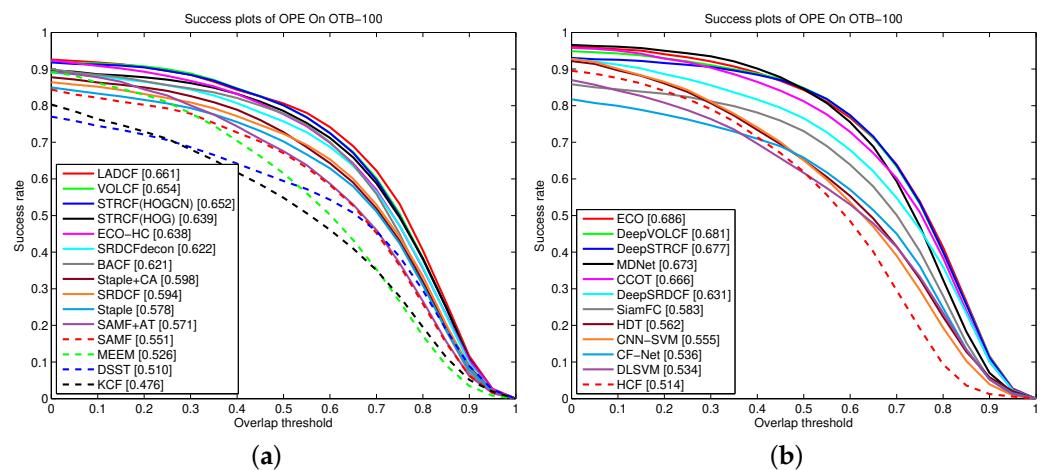


**Figure 7.** Comparison of the overlap success plots with those of state-of-the-art trackers on the OTB100 dataset: (a) trackers with HOG features and (b) trackers with deep features.



**Figure 8.** Qualitative evaluation of six video sequences (i.e., CarScale, Dog, boy, Human, Panda, and Trans) was performed. We show the results of VOLCF, STRCF, ECO-HC, BACF, SRDCF and SRDCFDecon with different colors.

#### 5.2.4. Temple-Colour Dataset

We performed VOLCF and DeepVOLCF experiments on the Temple-Colour dataset [58], which consists of 128 color sequences compared with the 9 state-of-the-art trackers mentioned in CVPR2013. Figure 9 shows a comparison of overlap success plots for all trackers with hog and deep features. While the performance of VOLCF is not as good as that of ECO [36], the score of VOLCF is the same as that of its counterpart STRCF [18] with deep features and the success plot of VOLCF with the HOG feature is 0.55% greater than that of STRCF.

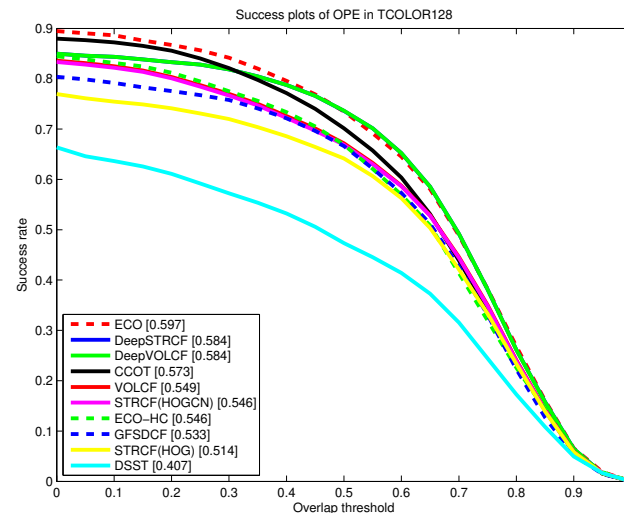


**Figure 9.** Results of the overlap success plot on Template-Colour. Here, we compare 10 trackers.

### 5.3. Analysis of Efficiency Experiment

Additionally, we also report the tracking speed (FPS) comparison on the OTB dataset in Tables 1 and 2.

It can be observed that our VOLCF tracker achieves 7.8 and 7.9 fps with hand-crafted features and deep features on OTB respectively. Compared to other state-of-the-art trackers using hand-crafted features, our proposed VOLCF surpasses the SRDCFdecon, SRDCF, and SAMF+AT methods in terms of fps and is on par with the GFSDCF (HOGCN) method. In terms of deep feature recognition, our VOLCF method outperforms most competing trackers, exceeding DeepSRDCF by 97.46%, slightly below CF-Net.

**Table 1.** The FPS results of trackers with hand-crafted features on OTB.

|     | VOLCF  | STRCF(HOG) | STRCF(HOGCN) | ECO-HC | SRDCFdecon | BACF  | Staple+CA     | SRDCF |
|-----|--------|------------|--------------|--------|------------|-------|---------------|-------|
| FPS | 7.8    | 31.5       | 24.3         | 42     | 2.0        | 26.7  | 35.3          | 5.8   |
|     | Staple | SAMF+AT    | SAMF         | MEEM   | DSST       | LADCF | GFSDCF(HOGCN) | KCF   |
| FPS | 23.8   | 2.2        | 11.5         | 22.4   | 15.6       | 18.2  | 7.8           | 171.8 |

**Table 2.** The FPS results of trackers with deep features on OTB.

|     | VOLCF | ECO | DeepSTRCF | HDT | HCF  | DeepSRDCF | SiamFC | CF-Net |
|-----|-------|-----|-----------|-----|------|-----------|--------|--------|
| FPS | 7.9   | 9.8 | 5.3       | 2.7 | 10.2 | 0.2       | 12.6   | 8.7    |

### 5.4. Analysis of the VOT Dataset

For further evaluation, we performed experiments on three VOT datasets, namely, VOT2016, VOT2017 and VOT2018. We compared the VOLCF algorithm with other trackers, including STRCF, DeepSTRCF [18], Staple [28], DSST [30], SRDCF [17], SiamFC [38], MEEM [57], and KCF [14], with HOG and deep features. Unlike the OTB dataset, the experimental effect on the VOT dataset is reported in three metrics:

- Accuracy measures the average overlap ratio between the ground truth and the predicted bounding box by trackers.
- Robustness refers to computing the average number of tracking failures over the sequence, which represents the failure rate; this metric includes six attributes (i.e., camera motion, illumination change, occlusion, motion change, size change, and empty attribute).
- The expected average overlap (EAO) estimates the accuracy of the estimated bounding box after a certain number of frames are processed since initialization and averages the no-reset overlap of a tracker.

#### 5.4.1. VOT2016 Dataset

We report the results on VOT-2016 [27], which consists of 60 challenging videos, in Table 3. The best results are marked in red, and the second-best results are marked in green. As shown in Table 3, DeepVOLCF has the best performance among all the trackers and outperforms the second-best tracker by 0.74%, 4.79% and 2.2% in terms of EAO, accuracy and robustness, respectively. VOLCF has the second-best performance in terms of EAO, achieving a great improvement over its counterpart STRCF. In addition, DeepVOLCF also performs favorably against DeepSTRDCF [18] by a gain of 8.6% in the EAO metric.

**Table 3.** Comparison with state-of-the-art trackers on the VOT-2016 dataset.

|             | VOLCF  | DeepVOLCF | DSST   | Staple | MDNet-N | SRDCF  | DeepSRDCF | STRCF | DeepSTRCF |
|-------------|--------|-----------|--------|--------|---------|--------|-----------|-------|-----------|
| EAO↑        | 0.3376 | 0.3401    | 0.1811 | 0.2952 | 0.2572  | 0.2471 | 0.2749    | 0.279 | 0.313     |
| Accuracy↑   | 0.5398 | 0.5729    | 0.537  | 0.5467 | 0.5421  | 0.5364 | 0.5294    | 0.53  | 0.53      |
| Robustness↓ | 1.37   | 0.9       | 2.52   | 1.35   | 1.2     | 1.5    | 1.17      | 1.32  | 0.92      |

#### 5.4.2. VOT2017 Dataset

The VOT2017 dataset contains 60 challenging sequences (some simple sequences are replaced with more difficult sequences in VOT2016) and has more accurate ground truths. We report the evaluation results on the VOT2017 benchmark in comparison with several state-of-the-art trackers (i.e., STRCF, DeepSTRCF [18], Staple [28], DSST [30], SRDCF [17], SiamFC [38], and MEEM [57]) and add failure metrics compared to VOT2016. The results are summarized in Table 4. The top three are marked in red, blue, and green, respectively. In terms of all the metrics, the performance of DeepVOLCF is much better than that of its counterpart DeepSTRCF [18], and it achieves the best performance among the trackers in terms of all the metrics, with the exception of accuracy.

**Table 4.** The results of VOLCF with the state-of-the-art trackers on VOT-2017.

|             | VOLCF   | DeepVOLCF | STRCF   | DeepSTRCF | Staple  | DSST    | SRDCF   | SiamFC  | MEEM    |
|-------------|---------|-----------|---------|-----------|---------|---------|---------|---------|---------|
| EAO↑        | 0.1718  | 0.2272    | 0.1729  | 0.1680    | 0.1694  | 0.0788  | 0.1189  | 0.188   | 0.1925  |
| Accuracy↑   | 0.4961  | 0.5212    | 0.4943  | 0.5024    | 0.5278  | 0.3969  | 0.4867  | 0.501   | 0.4623  |
| Robustness↓ | 2.467   | 1.533     | 2.4     | 2.333     | 2.45    | 5.167   | 3.467   | 2.083   | 1.9     |
| Failures↓   | 46.8525 | 27.6141   | 45.4806 | 41.8108   | 44.0194 | 95.5598 | 64.1163 | 34.0259 | 33.6046 |

To better demonstrate the superior performance of deepVOLCF, Figure 10 shows the accuracy–robustness (A-R) plots of the trackers. We compare the A-R plots of VOLCF and DeepVOLCF with those of seven trackers with different graphic symbols in Figure 10. It is noticeable that the best performance of the tracker would result in a closer position at the top-right corner of the plot. Thus, DeepVOLCF shows the best tracking effect for its top-right position, which is better than that of STRCF.

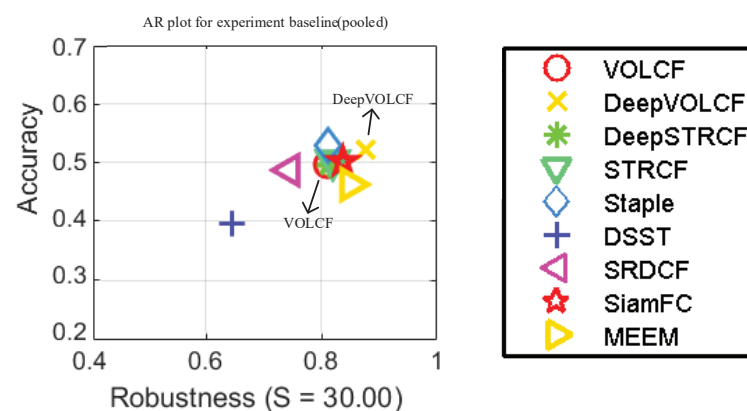


**Figure 10.** Accuracy–robustness (A-R) plots of 9 trackers on VOT2017. At the bottom right of the picture is the tracker represented by the logo graphic.

### 5.4.3. VOT2018 Dataset

The performance of our VOLCF tracker was also evaluated on VOT2018 for comparisons with 6 trackers, namely, STRCF [18], Staple [28], DSST [30], SRDCF [17], SiamFC [38] and KCF [14]. The VOT2018 dataset consists of 60 videos, and the results in terms of the EAO, accuracy, robustness and failure are presented in Table 5. VOLCF outperforms STRCF in terms of EAO, robustness and failure. In addition, DeepSTRCF obtained the highest scores, 0.1893 and 2.083 for EAO and robustness, respectively. Figure 11 shows a comparison of EAO at the experimental baseline among all the trackers. In contrast, DeepVOLCF is in the farthest right position among all the trackers, which means that it has the best accuracy, and VOLCF is in the third position among the EAO metrics.

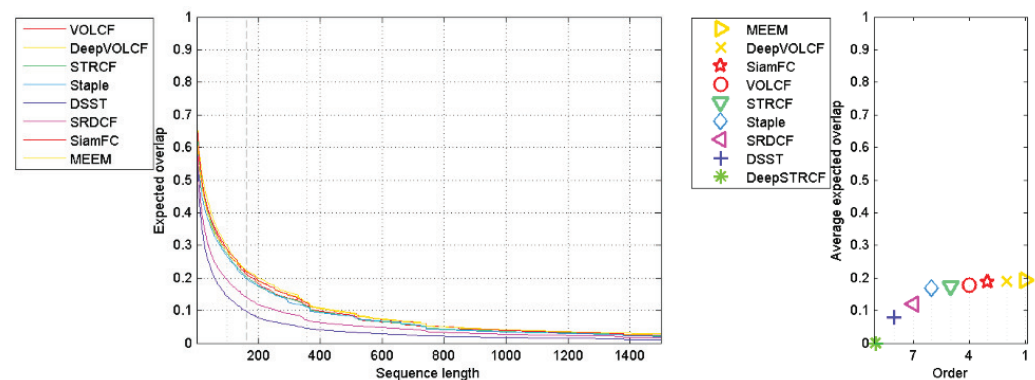


**Figure 11.** Expected averaged overlap performance for the baseline on VOT2018. A larger (right) value indicates better performance.

**Table 5.** Results of VOLCF with the state-of-the-art trackers on VOT-2018.

|             | VOLCF   | DeepVOLCF | STRCF   | Staple  | DSST    | SRDCF   | SiamFC  | KCF     |
|-------------|---------|-----------|---------|---------|---------|---------|---------|---------|
| EAO↑        | 0.178   | 0.1893    | 0.1731  | 0.1688  | 0.0793  | 0.1179  | 0.1875  | 0.1351  |
| Accuracy↑   | 0.4932  | 0.4883    | 0.4943  | 0.5278  | 0.3969  | 0.4867  | 0.501   | 0.448   |
| Robustness↓ | 2.167   | 2.083     | 2.322   | 2.45    | 5.167   | 3.467   | 2.083   | 2.75    |
| Failures↓   | 40.1876 | 39.5321   | 45.4806 | 44.0194 | 95.5587 | 64.1163 | 34.0259 | 50.0994 |

### 5.5. Analysis on the DTB70 Dataset

To better demonstrate the superiority of our model in tracking performance, we selected four video sequences from the DTB70 dataset to showcase the performance of our method in this article. DTB70 contains 70 video sequences. This dataset covers various challenging scenarios, including fast motion, scale variations, occlusions, blurriness, and changes in lighting conditions. In our experiments, the VOLCF model exhibited outstanding tracking performance on the DTB70 dataset. Compared to the current state-of-the-art trackers, including LADCF [15], SRDCF, DeepSRDCF [17], STRCF, DeepSTRCF [18], our approach performed excellently across multiple metrics in Table 6. LADCF suffers from severe tracking drift, leading to tracking failure, while STRCF also has issues with expanding the tracking area in the Bike video. In summary, our method consistently maintains good tracking performance, with the tracking frame always surrounding the target object. Particularly in handling small targets and complex backgrounds, the VOLCF model effectively maintained target tracking, even in cases of high similarity between the target and background, thereby avoiding template drift and model degradation.

**Table 6.** Comparison with state-of-the-art trackers on the DTB70 dataset.

|             | VOLCF  | DeepVOLCF | LADCF  | SRDCF  | DeepSRDCF | STRCF | DeepSTRCF |
|-------------|--------|-----------|--------|--------|-----------|-------|-----------|
| EAO↑        | 0.3056 | 0.3201    | 0.1972 | 0.2461 | 0.2766    | 0.259 | 0.288     |
| Accuracy↑   | 0.6308 | 0.5979    | 0.5012 | 0.5264 | 0.5664    | 0.587 | 0.5003    |
| Robustness↓ | 1.2    | 0.85      | 1.6    | 1.4    | 1.0       | 1.46  | 0.91      |

### 5.6. Analysis of Ablation Experiments

During the experiment, we found that the parameter of the loss function has an important effect on the performance. We validated the effectiveness of VOLCF by comparing existing trackers with variants of VOLCF. Here, the variants of VOLCF are obtained by changing the parameter of Equation (11) to obtain  $\Phi$ . Using the experiment on OTB100 as an example, we investigate the impacts of the model confidence term on the VOLCF. The results on the overlap success plot of different VOLCF variants and ECO-HC [36], STRCF [18], and SRDCF [17] are shown in Figure 12 based on HOG features and deep features. As shown in Figure 12, different parameters yield various experimental results. As a result, the VOLCF algorithm with the HOG feature outperforms the STRCF algorithm when the parameter is smaller than 0.01. The best-performing variant of DeepVOLCF ranks second place on OTB100.

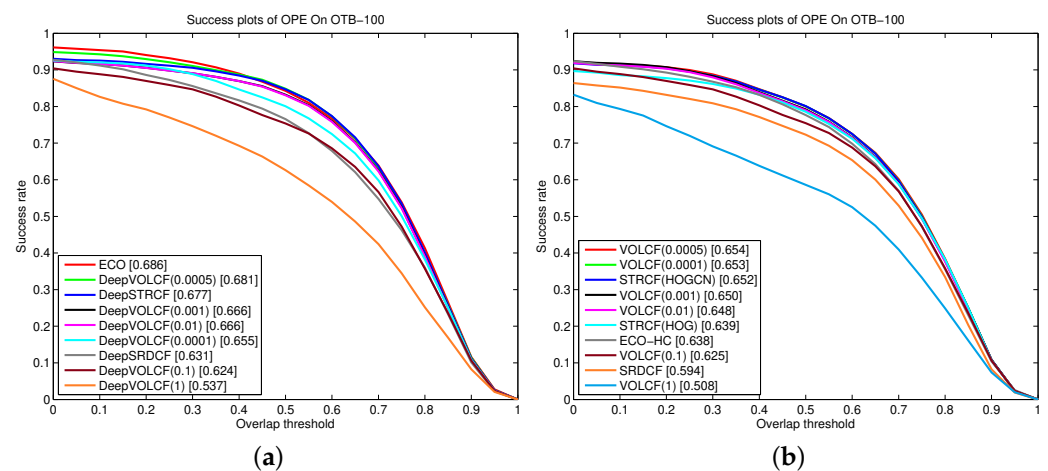


**Figure 12.** The overlap success plot of different VOLCF variants with (a) ECO-HC, (b) STRCF and SRDCF based on HOG features and deep features.

## 6. Conclusions

In this paper, we propose a variational online learning correlation filter (VOLCF) for visual tracking. The proposed method is obtained by introducing KL divergence and model confidence terms to solve the above problem from a variational online learning perspective. The KL divergence and model confidence terms improve the accuracy and robustness of the experiment and enhance the spatial-temporal consistency and alignment of the filter learning measurements. In addition, the ADMM method is adopted to optimize the proposed model by dividing the loss function into several subproblems. The extensive experimental results are compared with those of state-of-the-art trackers on the OTB, Temple-Colour, and VOT benchmark datasets with the HOG and deep features, demonstrating the effectiveness and robustness of our method.

**Author Contributions:** Methodology, Z.W.; Validation, Z.W.; Resources, L.D.; Data curation, F.L.; Writing—original draft, Z.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China (No. 62177029 and 62072256) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX21\_0740), China.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Kumar, A. *Visual Object Tracking Using Deep Learning*; CRC Press: Boca Raton, FL, USA, 2023.
2. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. Hift: Hierarchical feature transformer for aerial tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, New Orleans, LA, USA, 18–24 June 2021; pp. 15457–15466.
3. Yang, J.; Ge, H.; Yang, J.; Tong, Y.; Su, S. Online multi-object tracking using multi-function integration and tracking simulation training. *Appl. Intell.* **2022**, *52*, 1268–1288. [\[CrossRef\]](#)
4. Fang, Y.; Huang, Z.; Pu, J.; Zhang, J. AUV position tracking and trajectory control based on fast-deployed deep reinforcement learning method. *Ocean. Eng.* **2022**, *245*, 110452. [\[CrossRef\]](#)
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
6. Cao, Y.; Cao, J.; Zhou, Z.; Liu, Z. Aircraft Track Anomaly Detection Based on MOD-Bi-LSTM. *Electronics* **2021**, *10*, 1007. [\[CrossRef\]](#)
7. Zhang, L.; Fan, H. Visual object tracking: Progress, challenge, and future. *Innovation* **2023**, *4*, 100402. [\[CrossRef\]](#)
8. Amosa, T.I.; Sebastian, P.; Izhar, L.I.; Ibrahim, O.; Ayinla, L.S.; Bahashwan, A.A.; Bala, A.; Samaila, Y.A. Multi-camera multi-object tracking: A review of current trends and future advances. *Neurocomputing* **2023**, *552*, 126558. [\[CrossRef\]](#)
9. Lee, M.F.R.; Chen, Y.C. Artificial Intelligence Based Object Detection and Tracking for a Small Underwater Robot. *Processes* **2023**, *11*, 312. [\[CrossRef\]](#)
10. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Yu, Q.; Dinh, T.B.; Medioni, G. Online tracking and reacquisition using co-trained generative and discriminative trackers. In Proceedings of the European Conference on Computer Vision (ECCV), Marseille, France, 12–18 October 2008; pp. 678–691.
12. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
13. Henriques, J.F.; Rui, C.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the 12th European conference on Computer Vision, Florence, Italy, 7–13 October 2012; Volume 7575, pp. 702–715.
14. 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. [\[CrossRef\]](#)
15. Xu, T.; Feng, Z.; Wu, X.; Kittler, J. Learning Adaptive Discriminative Correlation Filters via Temporal Consistency Preserving Spatial Feature Selection for Robust Visual Object Tracking. *IEEE Trans. Image Process.* **2019**, *28*, 5596–5609. [\[CrossRef\]](#)
16. Gao, L.; Liu, B.; Fu, P.; Xu, M.; Li, J. Visual tracking via dynamic saliency discriminative correlation filter. *Appl. Intell.* **2022**, *52*, 5897–5911. [\[CrossRef\]](#)
17. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4310–4318. [\[CrossRef\]](#)
18. Li, F.; Tian, C.; Zuo, W.; Zhang, L.; Yan, M.H. Learning spatial-temporal Regularized Correlation Filters for Visual Tracking. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4904–4913. [\[CrossRef\]](#)
19. Cao, J.; Pang, J.; Weng, X.; Khirondkar, R.; Kitani, K. Observation-centric sort: Rethinking sort for robust multi-object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 9686–9696.
20. Wu, Y.; Sheng, H.; Zhang, Y.; Wang, S.; Xiong, Z.; Ke, W. Hybrid motion model for multiple object tracking in mobile devices. *IEEE Internet Things J.* **2022**, *10*, 4735–4748. [\[CrossRef\]](#)
21. Jayaraman, D.; Grauman, K. Slow and Steady Feature Analysis: Higher Order Temporal Coherence in Video. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3852–3861. [\[CrossRef\]](#)
22. Wiskott, L.; Sejnowski, T.J. Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Comput.* **2002**, *14*, 715–770. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Gui, J.; Sun, Z.; Wen, Y.; Tao, D.; Ye, J. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *arXiv* **2020**, arXiv:2001.06937.
24. Tyagi, K.; Rane, C.; Manry, M. Automated Sizing and Training of Efficient Deep Autoencoders using Second Order Algorithms. *arXiv* **2023**, arXiv:2308.06221.
25. Danelljan, M.; Van Gool, L.; Timofte, R. Probabilistic Regression for Visual Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7181–7190.
26. Wu, Y.; Lim, J.; Yang, M. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Cehovin, L.; Fernandez, G.; Vojir, T.; Hager, G.; Nebel, G.; Pflugfelder, R. The Visual Object Tracking VOT2016 Challenge Results. In Proceedings of the Computer Vision—ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 October 2016; Lecture Notes in Computer Science; Volume 9914, pp. 777–823. [\[CrossRef\]](#)

28. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H.S. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
29. Li, Y.; Zhu, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In Proceedings of the Computer Vision—ECCV 2014 Workshops, Zurich, Switzerland, 6–12 September 2014; Lecture Notes in Computer Science; Volume 8926, pp. 254–265.
30. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Accurate Scale Estimation for Robust Visual Tracking. In Proceedings of the British Machine Vision Conference 2014, Nottingham, UK, 1–5 September 2014; pp. 1–5.
31. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Discriminative Scale Space Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1561–1575. [[CrossRef](#)]
32. Xu, T.; Feng, Z.; Wu, X.; Kittler, J. Joint Group Feature Selection and Discriminative Filter Learning for Robust Visual Object Tracking. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7949–7959.
33. Galoogahi, H.K.; Fagg, A.; Lucey, S. Learning Background-Aware Correlation Filters for Visual Tracking. *arXiv* **2017**, arXiv:1703.04590.
34. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Adaptive Decontamination of the Training Set: A Unified Formulation for Discriminative Visual Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1430–1438.
35. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 8–10 October 2016; Lecture Notes in Computer Science; Volume 9909, pp. 472–488.
36. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939.
37. Han, Y.; Deng, C.; Zhao, B.; Zhao, B. spatial-temporal Context-Aware Tracking. *IEEE Signal Process. Lett.* **2019**, *26*, 500–504. [[CrossRef](#)]
38. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the Computer Vision—ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 October 2016; Lecture Notes in Computer Science; Volume 9914, pp. 850–865.
39. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H.S. End-to-End Representation Learning for Correlation Filter Based Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008.
40. Qi, Y.; Zhang, S.; Qin, L.; Yao, H.; Huang, Q.; Lim, J.; Yang, M. Hedged Deep Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; Volume 2, pp. 4303–4311.
41. Ma, C.; Huang, J.; Yang, X.; Yang, M. Hierarchical Convolutional Features for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3074–3082.
42. Qi, Y.; Zhang, S.; Qin, L.; Huang, Q.; Yao, H.; Lim, J.; Yang, M. Hedging Deep Features for Visual Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1116–1130. [[CrossRef](#)]
43. Du, F.; Liu, P.; Zhao, W.; Tang, X. Joint Channel Reliability and Correlation Filters Learning for Visual Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 1625–1638. [[CrossRef](#)]
44. Huang, Y.; Li, X.; Lu, R.; Qi, N. RGB-T object tracking via sparse response-consistency discriminative correlation filters. *Infrared Phys. Technol.* **2023**, *128*, 104509. [[CrossRef](#)]
45. Sun, C.; Wang, D.; Lu, H.; Yang, M. Learning Spatial-Aware Regressions for Visual Tracking. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8962–8970.
46. Wang, Y.; Kitani, K.; Weng, X. Joint object detection and multi-object tracking with graph neural networks. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 13708–13715.
47. Yang, X.; Li, S.; Ma, J.; Yang, J.y.; Yan, J. Co-saliency-regularized correlation filter for object tracking. *Signal Process. Image Commun.* **2022**, *103*, 116655. [[CrossRef](#)]
48. Cheng, X.; Zhang, Y.; Cui, J.; Zhou, L. Object Tracking via Temporal Consistency Dictionary Learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 628–638. [[CrossRef](#)]
49. Pu, L.; Feng, X.; Hou, Z. Learning Temporal Regularized Correlation Filter Tracker with Spatial Reliable Constraint. *IEEE Access* **2019**, *7*, 81441–81450. [[CrossRef](#)]
50. Guo, H.; Li, H.; Sun, N.; Ren, Q.; Zhang, A.; Wang, W. Concept drift detection and accelerated convergence of online learning. *Knowl. Inf. Syst.* **2023**, *65*, 1005–1043. [[CrossRef](#)]
51. Chen, Z.; Sheng, V.; Edwards, A.; Zhang, K. An effective cost-sensitive sparse online learning framework for imbalanced streaming data classification and its application to online anomaly detection. *Knowl. Inf. Syst.* **2023**, *65*, 59–87. [[CrossRef](#)]
52. Zhang, Y. Learning Label Correlations for Multi-Label Online Passive Aggressive Classification Algorithm. *Wuhan Univ. J. Nat. Sci.* **2024**, *29*, 51–58. [[CrossRef](#)]
53. Eckstein, J.; Bertsekas, D.P. On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **1992**, *5*, 293–318. [[CrossRef](#)]

54. Cehovin, L.; Kristan, M.; Leonardis, A. Is my new tracker really better than yours? In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, 24–26 March 2014; pp. 540–547.
55. Mueller, M.; Smith, N.; Ghanem, B. Context-Aware Correlation Filter Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1387–1395.
56. Bibi, A.; Mueller, M.; Ghanem, B. Target Response Adaptation for Correlation Filter Tracking. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 8–10 October 2016; Lecture Notes in Computer Science; Volume 9910, pp. 419–433.
57. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Lecture Notes in Computer Science; Volume 8694, pp. 188–203.
58. Liang, P.; Blasch, E.; Ling, H. Encoding Color Information for Visual Tracking: Algorithms and Benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5630–5644. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.