IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 2, 2018, accepted September 7, 2018, date of publication September 13, 2018,
date of current version October 8, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869766

Occlusion-Aware Correlation Particle Filter Target Tracking Based on RGBD Data

YAYU ZHAI, PING SONG, (Member, IEEE), ZONGLEI MOU, XIAOXIAO CHEN, AND XIONGJUN LIU
Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, Beijing Institute of Technology, Beijing 100081, China
Corresponding author: Ping Song (sping2002@bit.edu.cn)

This work was supported by the National Defense Basic Scientific Research Program of China under Grant JCKY2017602B012.

ABSTRACT In recent decades, there have been considerable improvements in target-tracking algorithms. However, aspects such as target occlusion, scale variation, and illumination changes still present significant challenges to existing algorithms. In this paper, we describe an occlusion-aware correlation particle filter target-tracking method based on RGBD data. First, we derive a target occlusion judgment mechanism based on a depth image and the histogram of oriented gradients (HOG) feature. We then formulate the tracking mechanism for the target prediction–tracking–optimization–redetection process using a correlation maximum likelihood estimation particle filter algorithm. We propose an adaptive update strategy whereby the system saves a well-tracked model when no occlusion occurs, and then uses this saved model to replace poorly tracked models in the event of occlusion. Furthermore, we consider the scale variation and adjust the target size according to the depth image, but we leave the HOG feature vector dimension of the target area unchanged. Thus, the problems such as model offset, scale variation, and loss of features are corrected over time. The experimental results demonstrate that the proposed target-tracking algorithm can detect target occlusion and track targets well, requires fewer calculations to perform target prediction–tracking–optimization–redetection, reduces the impact of illumination changes, and achieves better real-time performance and accuracy than many existing algorithms.

INDEX TERMS Adaptive update strategy, correlation maximum likelihood estimation particle filter, occlusion-aware, RGBD, target tracking.

I. INTRODUCTION

Target tracking has been the focus of considerable research in the field of machine vision. Target tracking entails establishing the position of the tracked object within a continuous video sequence and obtain its complete trajectory. The real-time, robust tracking of moving objects in a dynamic environment is an important part of target tracking. With the continuous improvement of computer technology, target tracking is increasingly used in fields including robotics, intelligent security, intelligent transportation, artificial intelligence, and other fields. Recently, the introduction of machine learning has significantly improved target-tracking technologies; however, there are still many challenges [1], [2], including scale changes, illumination changes, target deformation, scale variation, fast motion, target occlusion, and tracking speed.

Target-tracking methods can generally be divided into those based on a production model and those based on a discriminative model. Methods based on a production model determine the closest sample to the target model to be the current target state estimation. These methods typically proceed by calculating the joint probability of the target and the sample, and representative methods include the mean shift algorithm, the Kalman filter, the optical flow algorithm based on feature points, and the particle filter. The mean shift algorithm [3], [4] is a tracking method based on the distribution of the probability density. The search for the target always follows the direction of increasing probability gradient, and the iteration converges to the local peak of the probability density distribution. In [3], the mean shift algorithm models a target with nonparametric distributions of color features and locates the object with mode shifts. The Kalman filter algorithm [5] models the target's motion, uses an adaptive filtering method to estimate the process noise variance and measure noise variance, and then estimates the target's position in the next frame. The optical flow tracking method based on feature points [6], [7] extracts some feature points from the target, and then calculates their optical flow matching points in the next frame. This allows for a statistical determination of the target's position. The optical flow algorithm essentially uses feature points to characterize the target model. These production model methods are generally fast, have the ability to process occlusion problems, and can manage nonlinear systems. However, they all have problems with robustness and low accuracy.

Methods based on a discriminative model perform tracking using classification. The tracked target is considered to be the foreground and an online or offline training detector is used to distinguish the foreground target from the background. Representative methods include tracking–learning–detection (TLD), structured output tracking with kernels (Struck), differentiated-scale spatial tracking (DSST), circulant structure of tracking-by-detection with kernels (CSK), and the kernel correlation filter (KCF). TLD [8] is a novel tracking framework that explicitly decomposes the long-term tracking task into tracking, learning, and detection. Struck [9] performs adaptive visual target tracking based on structure output predictions. By explicitly introducing the output space to satisfy the tracking function, it can avoid intermediate classification links and output the tracking results directly. Furthermore, the algorithm uses a threshold mechanism to prevent the excessive growth of the support vector during the tracking process, thus ensuring real-time performance. Bolme *et al.* [10] proposed a method of learning the minimum output sum of squared error (MOSSE) correlation filter on grayscale images. This was the earliest use of correlation filters in target tracking. DSST [11] solves the problem of target scale changes in the tracking process based on the MOSSE tracking algorithm. Henriques *et al.* [12] proposed the CSK method, also based on MOSSE, which uses kernel tracking with a circulant matrix to solve the problem of dense sampling. Additionally, this method uses a Fourier transform to accelerate the detection process. In the context of big data, convolution features obtained using a deep learning training network model have been directly applied to the relevant filtering framework [13], resulting in better tracking at some additional computational expense. Methods based on the discriminative model offer high accuracy and robustness. However, in the event of target occlusion, their detection templates are incorrectly updated, resulting in a failure to follow the target.

The most difficult challenges in visual tracking are occlusion, scale variation, and tracking speed. Many advanced motion models have been devised to manage occlusions: parametric templates [14]–[16], linear Kalman filters [17], and nonlinear particle filters [18], [19] are the most useful approaches. In particular, Liu *et al.* [20] presented an initial target appearance model based on nonnegative matrix factorization (NMF) to describe the target's appearance along with an inverse NMF model where each learned base vector is regarded as a clustering center in a low-dimensional subspace. The model can produce more discriminative encoding vectors. Training a SVM classifier by the encoding vectors combined with using a particle filter can address the occlusion problem. Particle filters [21]–[25] have been applied to the problem of target occlusion using kernels, the combination of multiple features, and sparse coding representations. The common particle filter algorithm can solve the problems of partial occlusion and complete occlusion; however, its accuracy is reduced in the case of complex backgrounds and occlusion. The L1 minimization particle filter [25] can theoretically solve the problem of complex occlusion; however, Suha *et al.* [26] reported that the actual result is inadequate and does not consider the similarity calculation and template update problem. Meshgi *et al.* [27] proposed an occlusion-aware particle filter framework that maintains target tracking by predicting the appearance of occluded targets. The target template is updated by shifting relevant information and performs re-tracking by expanding the search area to find the occluded target. However, this method is slow because it entails retrieving the target from a large area. Henriques *et al.* [28] proposed the KCF tracking algorithm using the gradient histogram feature in the correlation filter tracking algorithm to improve CSK. This method can achieve fast target tracking, but cannot solve the problem of target occlusion. Yang *et al.* [29] presented a simple and fast method to improve the robustness of KCF by combining template and pixel-wise learners. Their method achieved a speed of 42 frames/s on large-scale challenging benchmark datasets, but still struggles effectively solve the target occlusion problem. Liu *et al.* [30] presented a sophisticated similarity metric termed "mutual buddies similarity" to exploit the relationship between multiple reciprocal nearest neighbors for target matching. They also designed a novel online template updating strategy named "memory filtering" to construct the current stable and expressive template by selecting a set of representative and reliable tracking results from the tracking history. This set of templates allows the algorithm to avoid poor tracking results caused by improper template updates following target occlusion.

In modern target tracking, the complexity of practical problems such as occlusion means that more nonlinear, non-Gaussian problems are being encountered; in addition, the tracking-speed requirements are ever-increasing. The particle filter method [18], [19] is widely used because of its excellent performance in nonlinear, non-Gaussian systems. Although the probability distribution given by the algorithm is only an approximation of the true distribution, the nonparametric characteristics of particle filters eliminate the need for the random variables to satisfy a Gaussian distribution when solving the nonlinear filtering problem. As a result, a wider range of distributions can be expressed with particle filters than with a Gaussian model. Particle filters also have stronger modeling capabilities for the nonlinear properties of variable parameters and can process occlusion problems in some extent. However, the main drawback is that particle filters require a large number of sample sizes to approximate the system's posterior probability density. The KCF tracking algorithm [28] uses a circulant matrix to acquire positive and negative samples with respect to the target, and applies ridge regression to train the target detector. The diagonalizable properties of the circulant matrix in the Fourier space are then used to successfully transform the matrix operations into the Hadamard product of the vector, which is the dot product of the elements. This greatly reduces the computational expense, allowing the algorithm to meet real-time requirements. However, this algorithm cannot manage the problem of complete target obstruction, and also lacks a truly adaptable model updating strategy.

To solve the above problems, we propose an occlusion-aware correlation particle filter (OACPF) target-tracking method based on RGBD data. The main contributions of the proposed approach are as follows:

1) We propose a target occlusion judgment mechanism based on a depth image and its histogram of oriented gradients (HOG) feature is proposed. We use an improved KCF method to fuse the target depth image and its HOG feature to accurately determine the target occlusion.
2) We combine the original KCF on HOG of RGB and the maximum likelihood estimation particle filter algorithm to form a target-tracking mechanism for prediction–tracking–optimization–redetection. In the case of no occlusion, KCF is used to track the target. The target particle is predicted and calibrated using the maximum likelihood estimation particle filter algorithm based on KCF. If occlusion occurs during the target tracking process, the target model stops updating. If the target appears after occlusion, the relevant maximum of the original model is retrieved by KCF around the target position predicted by the maximum likelihood particle filtering. This enables fast retrieval of the occluded target.
3) We propose a self-adaptive model update strategy. When no occlusion occurs, the well-tracked model is saved as a historical model. In the event of occlusion, the historical model is used instead of the poorly tracked model, thereby correcting the tracking over time. Furthermore, our method judges scale variation and adjusts the target size according to the depth image, but the HOG feature vector dimension of the target area remains unchanged. Problems such as model offset, scale variation, and loss of features are thus corrected over time.

Compared with existing target tracking algorithms, experimental results show that the proposed algorithm offers better performance in terms of accuracy, computation speed, and real-time performance.

II. OCCLUSION-AWARE CORRELATION PARTICLE FILTERING TARGET-TRACKING ALGORITHM BASED ON RGBD DATA

We propose an occlusion-aware correlation particle filter method for target tracking using RGBD data. First, we propose a target occlusion judgment mechanism based on a depth image and its HOG features. We then construct the tracking mechanism for target prediction–tracking–optimization–redetection using the correlation maximum likelihood estimation particle filter algorithm. In addition, we propose an adaptive update strategy that saves a well-tracked model when no occlusion occurs, and uses this model to replace a poorly tracked model in the event of occlusion. Furthermore, our method judges scale variation and adjusts the target size according to the depth image, but the HOG feature vector dimension of the target area remains unchanged. The overall process of the proposed method is shown in Algorithm 1.

A. TARGET OCCLUSION JUDGMENT MECHANISM BASED ON DEPTH IMAGE AND ITS HOG FEATURE

The main problem with respect to target occlusion is accurately determining whether occlusion is occurring. In this paper, we propose a target occlusion judgment mechanism based on a depth image and its HOG feature. We employ an improved KCF method to fuse the target depth image and its HOG feature to determine the occurrence of target occlusion.

1) DEPTH IMAGE HOG EXTRACTION

To calculate the depth image HOG feature, we divide the depth image into many cells and select different sliding windows according to practical considerations. We use a rectangular window for HOG feature extraction. The specific calculation steps are as follows:

(i) Calculate the gradient

We calculate the horizontal and vertical derivatives d_x and d_y from the convolution kernels $[-1, 0, 1]$, and $[-1, 0, 1]^{\mathrm{T}}$, respectively, to obtain the gradient information g for each pixel in each cell, that is,

$$g = \begin{bmatrix} d_x \\ d_y \end{bmatrix}. \tag{1}$$

Converting the gradient to polar coordinates, the direction β and size A of the gradient are obtained as

$$\beta = \arctan\left(\frac{d_x}{d_y}\right), \tag{2}$$

$$A = \sqrt{d_x^2 + d_y^2}. \tag{3}$$

(ii) Calculate the gradient histogram

We divide each image into multiple cells such that each cell includes $C \times C$ pixels. We calculate the β and A of each cell, and create an n-bin histogram for these cells, where each bin covers an angle of $360°/n$. The β and A in each cell are superimposed and assigned to each bin to obtain a gradient histogram of the cell, that is, a gradient vector.

(iii) Calculate the overall HOG feature vector

We compose a block of size $b \times b$ consisting of adjacent cells in the depth image. We normalize each block and obtain the size of the feature vector. Finally, we traverse the entire image using the sliding length S to obtain the feature vector

Algorithm 1 Proposed Occlusion-Aware Correlation Particle Filter Target Tracking Based on RGBD Data

```
Input: Image sequences and initialization
Output: Object tracking results
Initialize target template
Initialize N particles
for frame = 1:numel(image)
    if no occlusion or frame ==1
        Target occlusion judgment based on the depth image. (Eq. (11))
    end
    if no occlusion
        Target tracking using KCF on RGB data. (Eq. (22))
        Optimize particle filter parameters (Eq. (21))
        Update the target model (Section 2-C)
    end
    if occlusion
        Stop the target model update (Section 2-C)
        Predict and update particle filter model, and re-detect target using KCF on RGB data. (Section 2-B)
        if success
            Return to the state of no occlusion
        else
            Return to the state of occlusion
    end
end
```

of the entire image. The dimension is

$$Dim = (\frac{w - b \times c}{S} + 1) \times (\frac{h - b \times c}{S} + 1) \times b \times b \times bin, \tag{4}$$

where w and h denote the image width and height, respectively, and bin is the number of histogram slots.

For the HOG feature, we calculate the gradient vector is calculated by traversing the image using the sliding window, effectively representing the depth distribution of the image edge points within the contour.

2) IMPROVED KCF OCCLUSION JUDGMENT MECHANISM BASED ON TARGET DEPTH IMAGE AND ITS HOG FEATURE

The main idea of the KCF tracking algorithm [28] is that a discriminative correlation filter can be taught to target a new frame of the image. The algorithm uses the ridge regression method to model the problem. The training sample set is denoted as $\{(x_i, R_i)i = 1, \ldots, m\}$, where the samples x_i are characteristic representations of the image block and R_i denotes the sample labels assigned using a continuous Gaussian function. The kernel function $\kappa\,(x_1, x_2)$ maps x from a low-dimensional space to the high-dimensional $\varphi(x)$. The standard training process involves solving a linear regression function that minimizes the total residual $f(x) =< w, \varphi(x) >$. The ridge regression problem with the kernel function can then be expressed as

$$\min_{w} \sum_{i=1}^{m} (f(x_i) - R_i)^2 + \lambda ||w||^2, \tag{5}$$

where λ is the regularization coefficient and w represents the ridge regression model parameters. According to a theorem in [32], the solution to this problem can be expanded to a linear combination in the high-dimensional sample space $\varphi(x_i)$ satisfying $w = \sum_{i=1}^{m} \alpha_i \varphi(x_i)$, where α_i is the correlation coefficient. Substituting this formula into (5), the derivative can be obtained as

$$\alpha = (K + \lambda I)^{-1} R, \tag{6}$$

where α is the vector of α_i, R is the vector of R_i, I is the unit matrix, and K is the kernel matrix that satisfies $K_{ij} = \kappa\,(x_1, x_2)$.

If α has been obtained by training the sample set, then, in the detection of the current frame, the response of the regression function $f(z)$ for the newly input image block z can be expressed as

$$f(z) =< w, \varphi(z) >= (\sum_{i=1}^{m} \alpha_i \varphi(x_i))^T \varphi(z) = \sum_{i=1}^{m} \alpha_i \kappa(x_i, z). \tag{7}$$

In the implementation of the KCF tracking algorithm, the Gaussian kernel function is used to calculate

$$\kappa\left(x_i, x_j\right) = \exp(-\frac{1}{\sigma^2} ||x_i - x_j||^2). \tag{8}$$

Based on this, the properties of the circulant matrix can be applied to (6) and the regression coefficients obtained from training can quickly be calculated as

$$\hat{\alpha} = \frac{\hat{R}}{\hat{k}^{xx} + \lambda}, \tag{9}$$

where the symbol $\wedge$ represents the discrete Fourier transform of the corresponding vector. Applying the properties of the circulant matrix to (7), the regression response of the input feature to detect can be calculated as

$$\hat{f}(z) = diag(\hat{k}^{xz})\hat{\alpha} \Leftrightarrow f(z) = F^{-1}(diag(\hat{k}^{xz})\hat{\alpha}). \tag{10}$$

The point at which $f(z_i)$ is maximized gives the value of z_i that represents the detected position of the tracked target.

In this paper, we derive a new regression response function based on the depth image and its HOG feature to achieve the accurate detection of target occlusion. This function can be written as

$$f(z) = F^{-1}(diag(\hat{k}^{xz})\hat{\alpha}) - \varepsilon|d_l - d_p|, \tag{11}$$

where x and z of $\hat{k}^{xz}$ are the depth image HOG feature vectors of the training image block and the image block to be detected, respectively, ε is the depth image proportion coefficient, d_l is the mean of the depth image in the previous image block, and d_p is the mean of the image block in the depth image to be detected. ε was set according to the measurement range of the depth sensor. $f(z)$ represents the correlation between two adjacent depth frames. Specifically, we obtain $f(z)$ by calculating the correlation of the HOG features of two adjacent depth images and the difference in depth values between two adjacent depth images. We determine that the target is occluded when $f(z)$ is less than Fz or when the difference in $f(z)$ between two adjacent depth frames is greater than $\Delta(f(z))$.

B. TARGET PREDICTION–TRACKING–OPTIMIZATION–REDETECTION USING THE CORRELATION MAXIMUM LIKELIHOOD ESTIMATION PARTICLE FILTER ALGORITHM

In the tracking process, the original KCF using the HOG feature of the RGB image obtains the position of the target in the current frame according to the degree of correlation between the current frame and the image of the target in the previous frame. This does not involve motion state information about the target. When the target is completely obstructed, it becomes lost and the tracking result is degraded. Therefore, we propose a target-tracking algorithm using the correlation maximum likelihood estimation particle filter. When there is no occlusion, we use the original KCF algorithm to track the target, and incorporate the maximum likelihood estimation particle filter algorithm based on the original KCF to predict and calibrate the target position. If occlusion occurs during the target-tracking process, the target model stops updating; if the target appears after occlusion, the relevant maximum of the original model is retrieved by the KCF algorithm around the target position predicted by the maximum likelihood particle filter to complete the retrieval of the occluded target. The specific implementation process is as follows.

1) INITIALIZATION PHASE

The target tracking method based on particle filtering is a production tracking method [7]; thus, there must be an initial stage. For the first frame of the image, we manually select the target to be detected and extract the feature for the target area. In this paper, we use the RGB histogram as the target feature to establish the initial sample set through SIS sampling: $\left\{x_0^{(i)}, {}^1\!/_N\right\}_{i=1}^N$.

2) STATE TRANSITION PROCESS

The particle filter state transfer equation is

$$X_k = A \times X_{k-1} + r, \tag{12}$$

where r denotes noise and A denotes the state transition matrix. The state transition is performed according to (12), and the RGB template value of the new particle $\left\{x_k^{(i)}\right\}_{i=1}^N$ is calculated.

3) UPDATE THE PARTICLE WEIGHTS

We use the Bhattacharyya distance as a measure of the RGB histogram similarity between the target and the particle region. The Bhattacharyya coefficient for two consecutive distributions $p(u)$ and $q(u)$ is as follows:

$$\rho[p,q] = \sum_{i=1}^{N} \sqrt{p^{(i)}q^{(i)}}. \tag{13}$$

The target histogram in this paper is Q and the calculated RGB histogram of the particle region is py. The particle weight update formula is

$$w_k^{(i)} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1-\rho[Q,py]}{2\sigma^2}\right). \tag{14}$$

The weights can be normalized as

$$w_k^{(i)} = \frac{w_k^{(i)}}{\sum\limits_{i=1}^{N} w_k^{(i)}}. \tag{15}$$

Finally, we calculate the most likely position of the tracked target x_k at time k. We use the weighted average of all of the particles as the state estimate of the target location:

$$x_k = \sum_{i=1}^{N} x_k^{(i)} w_k^{(i)}. \tag{16}$$

4) UPDATE THE TARGET STATUS

The latest state of the target at time k is calculated as $x_k = \sum\limits_{i=1}^{N} x_k^{(i)} w_k^{(i)}$ using the latest set of particles and their weights.

5) RE-SAMPLING

An important problem with particle filtering is that, as the number of iterations increases, the weights become concentrated on a small number of particles. This leads to calculation errors and particles with smaller weights occupying excessive computing resources. Thus, we set a measure of the particle weights, $N_{eff} = 1 \Big/ \sum\limits_{i=1}^{N} \left(w_k^i\right)^2$. When the particle weights are below this threshold, the particles must be re-sampled. We reselect particles to continue computing according to the weight of the current particle. Unselected particles are abandoned. After re-sampling, all of the particles are assigned a weight of 1/N.

6) OPTIMIZATION

The actual state of the target and the predicted state of the particle filter are related as follows,

$$y_{k+1} = x_{k+1} + v, \tag{17}$$

where v is the error. To accurately predict the target position, we consider the variances σ_r^2, σ_w^2, and σ_v^2 to be equal, where σ_r^2, σ_w^2, and σ_v^2 are the variances corresponding to r, w, and v, respectively. We obtain a new equation based on the actual state of the target detected by the KCF algorithm using the RGB HOG feature and the predicted state obtained using particle filtering:

$$y_{k+1} = x_{k+1} + N(0, \sigma^2). \tag{18}$$

The actual state and predicted state of the target obey the Gaussian distribution, that is, $y_{1:K} \sim N\left(x_{1:K}, \sigma^2\right)$. According to the available maximum likelihood estimate,

$$l(\sigma | y_{1:K}, x_{1:K}) = -\frac{K \ln(2\pi\sigma^2)}{2} - \frac{\sum_{k=1}^{K} (y_k - x_k)^2}{2\sigma^2}. \tag{19}$$

By determining the partial derivative of σ^2 for the above formula and setting it to zero, we have

$$\sigma^2 = \frac{1}{K}\sum_{k=1}^{K} (y_k - x_k)^2. \tag{20}$$

To apply this method to online systems, the recursive relationship between σ_{k+1}^2 and σ_k^2 can be obtained as follows:

$$\sigma_{k+1}^2 = \frac{k-1}{k}\sigma_k^2 + \frac{k-1}{k}(y_k - x_k)^2. \tag{21}$$

During particle filter target tracking, the variances σ^2 and the particle distribution are adjusted recursively using the original KCF based on the HOG of RGB data; this can improve the speed and accuracy of target prediction.

7) OCCLUSION AND REDETECTION PROCESSING

When the target is occluded, the target model stops updating. When the target reappears after occlusion, we retrieve the relative maximum value of the original model by using the original KCF on the HOG of RGB data around the target position, as predicted by the maximum likelihood particle filter. According to the KCF method described above, the regression response function based on the KCF using the RGB HOG feature is obtained as

$$f(z) = F^{-1}(diag(\hat{k}^{xz})\hat{\alpha}), \tag{22}$$

where the x and z of $\hat{k}^{xz}$ are the RGB HOG feature vectors corresponding to the training image block and the image block to be detected, respectively. α is the correlation coefficient. The point at which $f(z)$ is greatest represents the position of the tracked target. The target retrieval after occlusion is considered successful when the difference in $f_c(z)$ between the last frame before occlusion and the next frame after occlusion is less than $\Delta\,(f_c(z))$.

C. ADAPTIVE MODEL UPDATE STRATEGY

The original KCF adopts a real-time update strategy to improve its adaptability. During the processing stage, the target model is updated at every frame to adapt to temporal changes in the target appearance characteristics. However, the target model is based on the current frame only, thereby ignoring the influence of the previous frame. The model loses the feature values pertaining to the target, thereby degrading the tracking results when the target becomes occluded or the scale changes.

In the proposed method, we update the feature models corresponding to the RGB of the target and depth image of the target based on the KCF of the RGB HOG feature and the depth image HOG feature, respectively. During the target tracking process, we update the historical model built from the well-tracked frames as a candidate model with a certain weight. When the target becomes occluded, the candidate model replaces the original model for tracking. The main steps are as follows:

Step 1: Initialize the candidate model, including the RGB image feature model and the depth image feature model;

Step 2: When the KCF based on the RGB HOG feature outputs a good tracking result, we update the two models according to the original KCF algorithm, and assign the result to the candidate model. During the update process, the target size is determined according to the depth image; if there is a certain degree of change, then we adjust the target size according to the depth image, but we do not change the HOG feature vector dimension of the target area. The models are updated according to the original KCF algorithm, and the candidate model is updated simultaneously.

Step 3: If the target is determined to be occluded using the improved KCF based on the target depth image and its HOG feature, the original model stops updating and the candidate model is used instead. The candidate model is used to re- detect the target once the target reappears.

III. EXPERIMENTAL RESULTS

We evaluated various aspects of the proposed OACPF algorithm experimentally. We compared OACPF with an RGBD SVM algorithm with an occlusion indicator (OI+SVM) [31], adaptive color-based particle filter (ACPF) [22], KCF [28], Background-Aware Correlation Filters (BACF) [33], Dual Linear Structured SVM (DLSSVM) [34], Scale-Adaptive Kernel Correlation Filter (SAMF) [35], Simple Tracker Combining Complementary Cues (STAPLE) [36], and RGBD single-object tracker with KCF tracker (DSKCF) [37].

All of the tracking algorithms were applied to five standard annotated video sequences from the Princeton Tracking dataset. Additionally, we further compared OACPF to the existing multiple-registration algorithms using 95 un-annotated test videos from the Princeton Tracking dataset.

A. DATA AND EVALUATION CRITERIA

The Princeton Tracking dataset consists of 100 RGBD video sequences acquired using the Microsoft Kinect. Of these, new_ex_occ4, face_occ_5, bear_front, child_no1, and zcup_move_1 are annotated, whereas the remaining 95 video sequences are un-annotated. In all of the video sequences, people or objects exhibit different degrees of movement, and some people or objects are occluded during the movement.

To verify the effectiveness of the algorithm, we use the following evaluation criteria: the center position (CP), the area under the curve (AUC), the center position error (CPE), false tracking (FT), mistaken tracking (MT), and frames per second (FPS). CP is mainly used to evaluate target occlusion detection and whether the target can be re-tracked after it reappears. AUC refers to the area under the tracking success rate curve for different overlapping thresholds. To provide better insight into the algorithm's outcomes, we define CPE as the L2-norm difference of the center position between the detected bounding box and actual bounding box. In the case of target occlusion, FT denotes that the algorithm cannot judge occlusion. MT refers to very little or no agreement between the target area detected by the algorithm and the actual target area.

B. TARGET OCCLUSION JUDGMENT RESULT BASED ON THE DEPTH IMAGE AND ITS HOG FEATURE

To validate the target occlusion judgment based on a depth image and its HOG feature, we used five annotated standard video sequences from the Princeton Tracking dataset. Three of these video sequences (bear_front, face_occ_5, and new_ex_occ4) contain occlusion problems. We use the CP judgment criteria to evaluate our method for these three video sequences. For the proposed OACPF algorithm, we set the number of initial particles in the maximum likelihood estimation to 100; According to the 0.5m to 5m measurement range of the Kinect, we set ε to 0.1 – 0.2, Fz to 0.2, $\Delta\,(f(z))$ to 0.3 – 0.5, and $\Delta\,(f_c(z))$ to 0.1 – 0.25.

We use several existing methods for comparison:

(1) ACPF is a particle filter that integrates color distributions and edge-based image features [22];

(2) OI+SVM uses a latent SVM with HOG features extracted from the RGB and depth images and an additional optical flow module to monitor occluding objects [31];

(3) KCF uses the circulant matrix of the target area to collect positive and negative samples, and applies ridge regression to train the target detector. A correlation filter is trained based on the information in the current and previous frame. The correlation is then calculated with the new input frame, and the resulting confidence map forms the predicted tracking result [28];

(4) BACF is a background-aware correlation filter based on hand-crafted features that can efficiently model how both the foreground and background of the object varies over time [33];

(5) DLSSVM uses a primal classifier update formula in which the learning step size is computed in closed form. An intersection kernel is used to provide feature representations, with an explicit feature map to improve tracking performance. The algorithm offers multi-scale estimation to address the drift problem [34];

(6) SAMF uses an effective scale adaptive scheme to tackle the problem of the fixed template size in the kernel correlation filter tracker. Features including HOG and color-naming are integrated to further boost the tracking performance [35];

(7) STAPLE is a simple tracker combining complementary cues in a ridge regression framework. This approach can handle illumination changes within a sequence [36];

(8) DSKCF is an RGB-D single-object tracker built on the extremely fast RGB-only KCF tracker [37].

The horizontal and vertical coordinates outputted by the different algorithms are shown in Fig. 1. NaN indicates that the occlusion cannot be judged. For the bear_front video, the frame locations corresponding to target occlusion were partially determined by OI+SVM (39, 57, NaN, NaN, NaN, NaN) and DSKCF (40, 72, 128, 148, 206, NaN), undetermined by ACPF, BACF, DLSSVM, SAMF, STAPLE, and KCF (NaN, NaN, NaN, NaN, NaN, NaN), and consistently determined by OACPF (42, 70, 127, 151, 207, 251). The actual frame locations were (41, 68, 125, 153, 205, 250). For the face_occ_5 video, the frame locations corresponding to target occlusion were judged as follows: OI+SVM (69), DSKCF (170), ACPF/ BACF/ DLSSVM/ SAMF/ STAPLE/KCF (NaN), and OACPF (169); the actual frame position was (168). For the new_ex_occ4 video, the frame locations corresponding to target occlusion were judged to be: OI+SVM (25), DSKCF (30), ACPF/ BACF/ DLSSVM/ SAMF/ STAPLE/KCF (NaN), and OACPF (28), while the actual frame position was (29).

This analysis and [33]–[36] indicate that ACPF, SAMF, BACF, DLSSVM, STAPLE, and KCF cannot accurately identify the occlusion problem when the target is occluded. DSKCF and OI+SVM showed some ability to judge occlusion. OACPF accurately judged single or multiple occlusions in all of the video sequences.

Table 1 presents the errors between the actual horizontal/vertical coordinates and those outputted by OI+SVM, ACPF, KCF, SAMF, BACF, DLSSVM, DSKCF, STAPLE, and OACPF. The experimental results show that OACPF achieved the smallest error for all three video sequences.

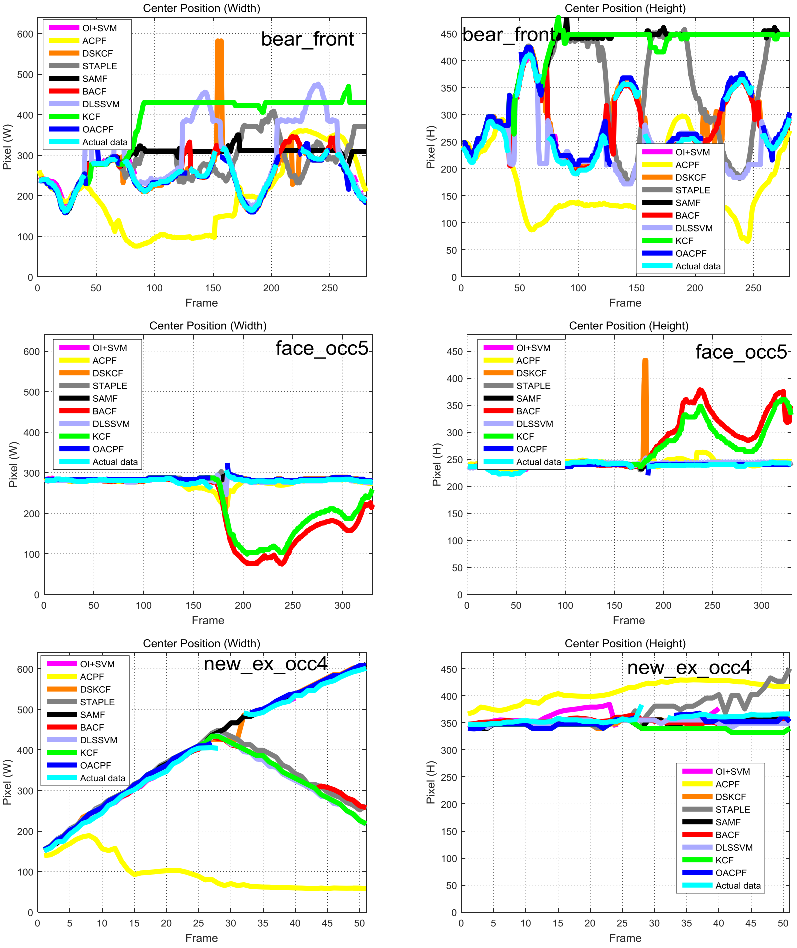


FIGURE 1. Horizontal and vertical coordinates outputted by OI+SVM, ACPF, KCF, SAMF, BACF, DLSSVM, DSKCF, STAPLE, and OACPF for bear_front, face_occ_5, and new_ex_occ4 video sequence tracking. The experimental results show that the OACPF method had the smallest error and highest occlusion accuracy.

TABLE 1. Errors between the actual horizontal/vertical coordinates and those obtained by OI+SVM, ACPF, KCF, SAMF, BACF, DLSSVM, DSKCF, STAPLE, and OACPF.

Algorithm	Horizontal/ Vertical error (bear_front)	Horizontal/ Vertical error (face_occ_5)	Horizontal/ Vertical error (new_ex_occ4)
OACPF	**19.1% / 21.06%**	**2% / 2.08%**	**6.92% / 6.37%**
OI+SVM	42.95% / 45.47%	5.74% / 6.7%	25.84% / 10.74%
ACPF	49.28% / 52.41%	5.72% / 6.58%	63.37% / 10.82%
KCF	82.77% / 79.36%	24.68% / 17.53%	31.99% / 10.28%
SAMF	42.84% / 90.44%	5.62% / 21.53%	9.33% / 40.81%
BACF	26.12% / 38%	28.45% / 22.09%	30.3% / 41.81%
DLSSVM	45.28% / 34.99%	5.2% / 21.08%	32.14% / 41.45%
DSKCF	26% / 38.79%	5.7% / 22.45%	9.07% / 40.71%
STAPLE	44.81% / 71.77%	5.79% / 21.11%	30.87% / 35.86%

C. VERIFICATION OF CORRELATION MAXIMUM LIKELIHOOD ESTIMATION PARTICLE FILTERING ALGORITHM

Here, we detail the results of our algorithm and the comparison algorithms with respect to the following evaluation metrics: AUC, CPE, FT, MT, and FPS.

Table 2 compares the tracking algorithms in terms of the AUC, CPE, FT, MT, and FPS results. All of the experiments were conducted on a 3.2 GHz 64-bit AMD Ryzen 1600 computer, with OACPF, OI+SVM, ACPF, KCF, SAMF, BACF, DLSSVM, DSKCF, and STAPLE implemented in Matlab. OACPF achieved the best scores with respect to AUC, CPE, FT, and MT. Although KCF achieved the highest FPS value, this algorithm performed poorly in other respects, particularly in the event of occlusion. OACPF achieved a superior FPS score to ACPF because the correlation maximum likelihood

TABLE 2. Comparison of the tracking algorithms OACPF, OI+SVM, ACPF, KCF, SAMF, BACF, DLSSVM, DSKCF, and STAPLE in terms of AUC, CPE, FT, MT, and FPS. Bold font denotes best score for each metric.

Algorithm	AUC	CPE	FT	MT	FPS
OACPF	**0.7369**	**11.3257**	**1.2**	**0.0**	17.014
OI+SVM	0.63478	21.2788	7.2	2.2	0.6
ACPF	0.26068	92.59406	12.6	35.0	14.259
KCF	0.55658	74.85834	12.6	33.0	**59.859**
BACF	0.7058	23.7446	12.6	4.4	18.48
DLSSVM	0.5887	37.2621	12.6	44.2	6.233
SAMF	0.6113	36.461	12.6	53	11.058
STAPLE	0.6045	53.4791	12.6	37	9.753
DSKCF	0.6957	11.5442	3.4	7.6	14.778

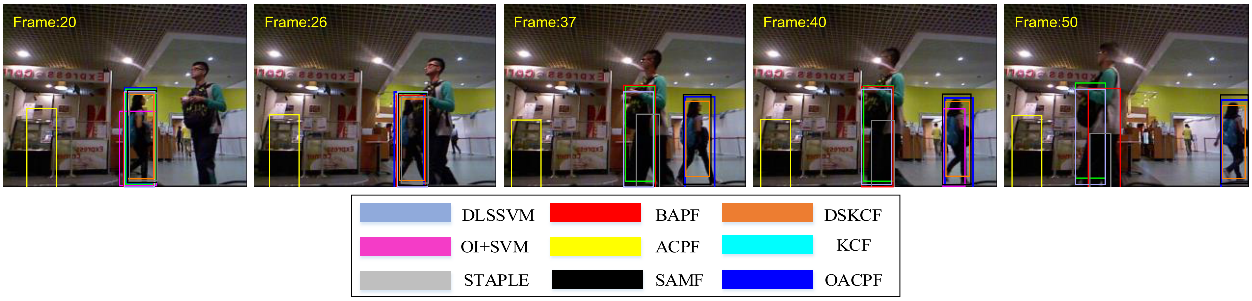


FIGURE 2. Tracking process of OI+SVM, ACPF, KCF, OACPF, SAMF, BACF, DLSSVM, DSKCF, and STAPLE algorithms on new_ex_occ4 video sequence.

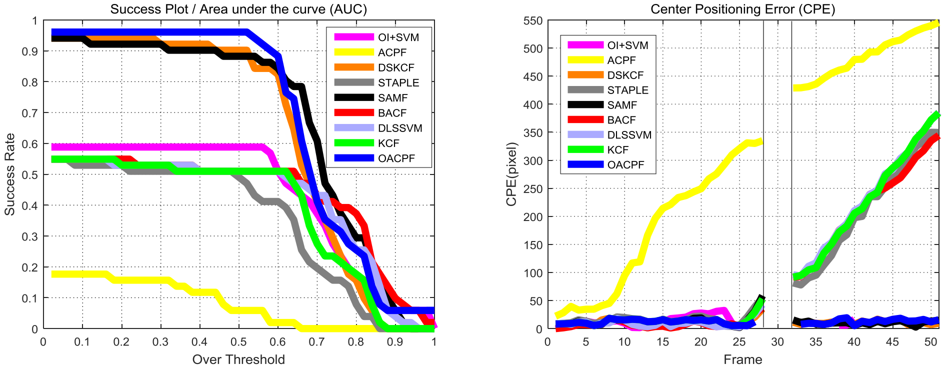


FIGURE 3. AUC and CPE for OACPF, OI+SVM, ACPF, KCF, SAMF, BACF, DLSSVM, DSKCF, and STAPLE when tracking the target in the new_ex_occ4 video sequence.

particle filter algorithm effectively reduces the number of particles as well as the amount of calculations, thereby improving the overall computation speed.

Figs. 2 and 3 provide a detailed illustration of the tracking process of the different algorithms on the new_ex_occ4 video sequence with respect to the AUC and CPE parameters. The algorithms did not output the tracking result when the tracking target was missing or occluded. This analysis shows that the ACPF algorithm produced significant tracking errors because of its vulnerability to the background. OI+SVM can judge target occlusion, but its sensitivity was too high, resulting in occlusion detection in the absence of occlusion. When the target reappeared, OI+SVM attempted to track the target again, but failed. KCF, BACF, DLSSVM, and STAPLE tracked the target well in the absence of target occlusion, but could not judge the occurrence of occlusion or re-track the target after it reappeared. SAMF tracked the target well in the absence of target occlusion; it cannot accurately judge the occlusion of the target, but could re-track the target once it reappeared. DSKCF judged partial occlusion and re-tracked the target fairly well. The AUC and CPE output curves demonstrate the superior tracking performance of OACPF, which has a small false tracking rate (the maximum success rate is close to 1). OACPF also recovered the target effectively after occlusion and had small CPE values.

Fig. 4 shows the distribution region of the particles in the OACPF algorithm and the search region of the KCF circulant matrix. The OACPF algorithm combined with the KCF detection results provides the basis for the prediction and update of the particle filtering, reducing the distribution area and the number of particles, thereby improving the overall speed of

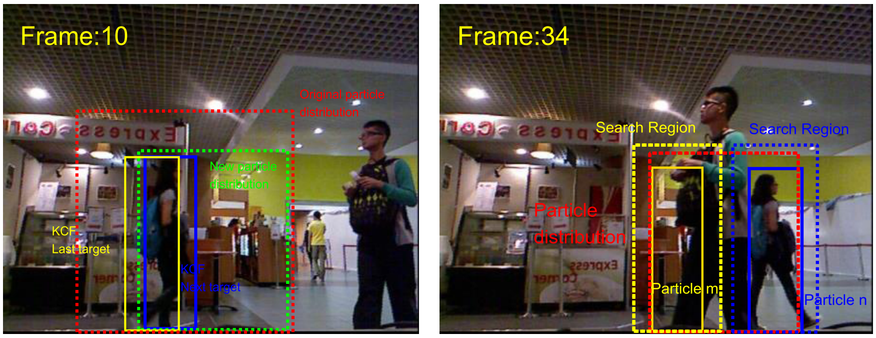


FIGURE 4. Particle distribution area in our OACPF algorithm and the search area of the KCF circulant matrix. OACPF reduces the number of particles and improves the speed of the overall algorithm. Particles in the distribution area provide possible locations for re-detection after the target becomes occluded, allowing the re-tracking of the target.

TABLE 3. Quantitative comparison of state-of-the-art algorithms for 95 test video sequences in the Princeton Tracking Dataset. Best results are shown in red, second and third best are marked in blue and green, respectively. Results were calculated by the online evaluation system provided by the dataset creators: http://tracking.cs.princeton.edu/eval.php.

Algorithm	Avg. Rank	Target Type			Target Size		movement		occlusion		Motion Type	
		Human	Animal	Rigid	Large	Small	Slow	Fast	Yes	No	Passive	Active
OAPF	3.27	0.64	0.85	0.77	0.73	0.73	0.85	0.68	0.64	0.85	0.78	0.71
3D-T	3.45	0.81	0.64	0.73	0.80	0.71	0.75	0.75	0.73	0.78	0.79	0.73
RGBDOcc+OF	3.82	0.74	0.63	0.78	0.78	0.70	0.76	0.72	0.72	0.75	0.82	0.70
DSKCF_SHAP	3.91	0.71	0.71	0.74	0.74	0.70	0.76	0.70	0.65	0.81	0.77	0.70
DS-KCF	6	0.67	0.61	0.76	0.69	0.70	0.75	0.67	0.63	0.78	0.79	0.66
OACPF	6.17	0.70	0.66	0.68	0.69	0.70	0.75	0.67	0.73	0.76	0.71	0.69
DSKCF-CPP	6.36	0.65	0.64	0.74	0.66	0.69	0.76	0.65	0.60	0.79	0.80	0.64
RGBD+OF	6.82	0.64	0.65	0.75	0.72	0.65	0.73	0.66	0.60	0.79	0.74	0.66
hiob_lc2	6.82	0.53	0.72	0.78	0.61	0.70	0.72	0.64	0.53	0.85	0.77	0.62
BBT	10.36	0.52	0.55	0.69	0.58	0.60	0.77	0.52	0.52	0.69	0.70	0.55
PCdet_flow	11	0.51	0.52	0.73	0.63	0.56	0.74	0.53	0.55	0.64	0.75	0.53
ASKCF-OHT	11.36	0.52	0.50	0.72	0.59	0.59	0.67	0.56	0.52	0.68	0.72	0.54
SAMF+Depth	14.45	0.45	0.50	0.67	0.52	0.55	0.65	0.49	0.41	0.72	0.66	0.49
LDP_SVT	15.36	0.47	0.59	0.56	0.53	0.52	0.56	0.51	0.41	0.68	0.58	0.50
KCF	15.45	0.42	0.50	0.65	0.48	0.55	0.65	0.47	0.41	0.68	0.65	0.47
RGBOF	15.64	0.47	0.47	0.64	0.47	0.58	0.57	0.52	0.47	0.62	0.63	0.49
LDPSTRUCK	15.82	0.46	0.59	0.54	0.52	0.52	0.56	0.50	0.40	0.68	0.56	0.50
PCdet	17.82	0.41	0.42	0.62	0.55	0.44	0.58	0.45	0.46	0.52	0.65	0.43
Dhog	18.64	0.43	0.48	0.56	0.47	0.50	0.53	0.47	0.38	0.64	0.54	0.47
Struck	20	0.35	0.47	0.53	0.45	0.44	0.58	0.39	0.30	0.64	0.54	0.41
VTD	20.55	0.31	0.49	0.54	0.39	0.46	0.57	0.37	0.28	0.63	0.55	0.38
RGB	22.45	0.27	0.41	0.55	0.32	0.46	0.51	0.36	0.35	0.47	0.56	0.34
CT	24.45	0.31	0.47	0.37	0.39	0.34	0.49	0.31	0.23	0.54	0.52	0.34
PCflow	24.55	0.35	0.29	0.44	0.42	0.33	0.47	0.33	0.32	0.43	0.51	0.35
TLD	24.91	0.29	0.35	0.44	0.32	0.38	0.52	0.30	0.34	0.39	0.50	0.31
MIL	25.18	0.32	0.37	0.38	0.37	0.35	0.46	0.31	0.26	0.49	0.40	0.34
SemiB	27.64	0.22	0.33	0.33	0.24	0.32	0.38	0.24	0.25	0.33	0.42	0.23
OF	29	0.18	0.11	0.23	0.20	0.17	0.18	0.19	0.16	0.22	0.23	0.17

the algorithm. Additionally, the particles in the distribution area provide possible locations for re-detection after the target becomes occluded. Once the target reappears, KCF completes the maximum matching of the original model according to the established search area, and achieves the re-tracking of the target.

Fig. 5 shows a visual review of the algorithms' performance on four other annotated sequences. As can be seen from the figure, OAPCF, KCF, SAMF, BACF, DLSSVM, DSKCF, and STAPLE effectively track the target when it is not occluded, whereas ACPF and OI+SVM exhibit partial tracking errors because of their vulnerability to

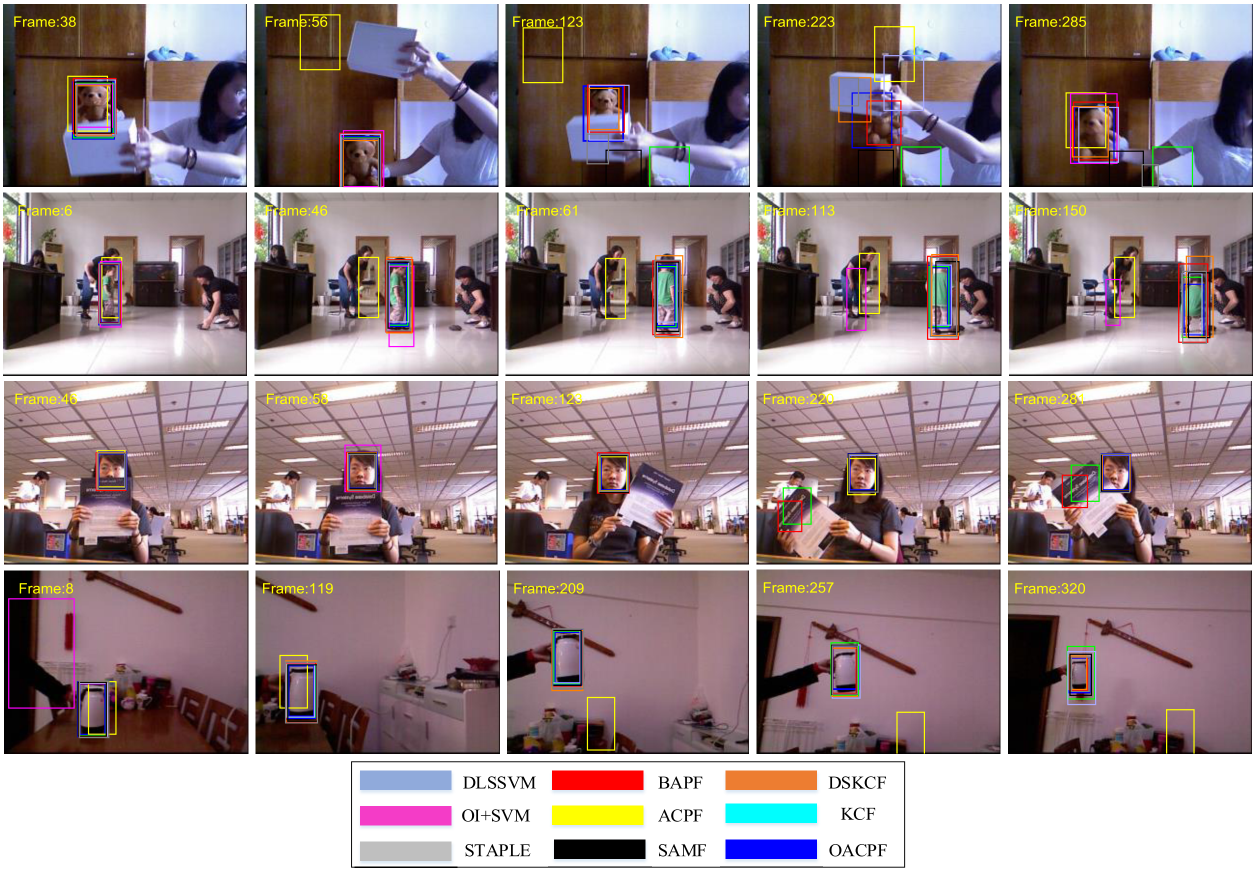


FIGURE 5. Visual review of the algorithms' performance on four other annotated sequences. These results show that OACPF performed favorably against OI+SVM, ACPF, KCF, SAMF, BACF, DLSSVM, DSKCF, and STAPLE.

the background. With the target occluded, KCF and ACPF cannot effectively judge the occlusion, resulting in subsequent tracking errors. SAMF, BACF, DLSSVM, and STAPLE cannot judge the occurrence of occlusion, but they can retrack the target to some extent. OACPF determines occlusion more accurately than DSKCF and OI+SVM, and successfully recovers target tracking after the occluded sequence.

To test the applicability of OACPF, we performed further experiments on 95 unannotated video sequences in the Princeton tracking dataset and evaluated the results using a tool on the Princeton tracking website (http://tracking.cs.princeton.edu/eval.php). The tracking results were compared with those obtained by a number of state-of-the-art RGB and RGBD algorithms including OAPF [27], DS-KCF [37], KCF [28], Struck [9], TLD [8], VTD [16], CT [38], MIL [39], SemiB [40], OF [41] and others. The experimental results are presented in Table 3.

The experimental results in Table 3 are classified according to target type (human, animal, rigid), target size (large or small), movement (slow or fast), occlusion (yes or no), and motion type (active or passive). The most accurate algorithm is marked in red, and the second and third most accurate are marked in blue and green, respectively. The average ranking across all of the classifications is presented in the first column.

OACPF exhibits good overall performance, ranking first in the case of target occlusion. Overall, the proposed OACPF method offers significant improvement over many existing algorithms. Compared with production model-based methods such as OAPF, the overall result of the proposed OACPF approach was relatively poor; however, OACPF performs better on human targets and in scenarios with occlusion. Additionally, according to [27], OAPF has a slower processing speed and is not suitable for real-time applications. Compared with methods based on the discriminative model, such as TLD, KCF, and Struck, the overall result and all of the aspects of the proposed OACPF approach are superior.

IV. DISCUSSION

The results presented in the previous section clearly demonstrate the effectiveness and efficiency of OACPF, with KCF and the recursive maximum likelihood estimation particle filters enhancing and complementing each other well. We highlight the following conclusions:

1) The target occlusion judgment mechanism based on a depth image and its HOG feature effectively addresses the problem of target occlusion. As shown in Fig. 1, compared with OI+SVM, ACPF, KCF, SAMF, BACF, DLSSVM, DSKCF, and STAPLE, the proposed OACPF effectively and accurately determine target occlusion.

2) Particle filters require dense particles for state transfer during target tracking. This process requires a great deal of time, which was reduced by using KCF. The KCF method can refine particles to cover target object states with recursive maximum likelihood estimation particle filters. Thus, the number of particles required in the target tracking process is reduced. As shown in Fig. 4, the actual state of the target was obtained during target tracking by KCF. According to the previous target and the new target, the maximum likelihood estimation method combines the actual state with the particle filter to provide a reliable basis for the prediction and update of the particle filter, and further redefines the distribution area of the particles as the new particle distribution. This method reduces the number of particles in a limited particle distribution area, thereby reducing the overall number of calculations and enabling real-time target tracking.

3) KCF cannot solve the problems of occlusion and scale variation; however, these are managed using particle filters. The particle filters make multiple assumptions and estimate the target position through state transitions, providing possible locations for the re-detection of the occluded target. Therefore, particle filters effectively help KCF to solve the occlusion problem, as demonstrated in Fig. 4 for the challenge of tracking an occluded object. As shown in Fig. 4, if the search area of KCF is only the search region of particle m, the tracking process fails after the target becomes occluded. However, searching for the target based on the position of the particle filter in the search region of particle n achieves re-tracking using the maximum value of KCF.

4) Adaptive update strategy: as shown in Fig. 2, when no occlusion occurs, the well-tracked model is saved as a historical model. After occlusion occurs, this historical model replaces the poorly tracked model. Furthermore, scale variations are judged according to the depth image. The target size is also adjusted according to the depth image, but the HOG feature vector dimension does not change. Problems such as model offset, scale variation, and loss of features are corrected over time. After the target reappears, the historical model is used to perform KCF maximum relevance matching, thereby completing the target's re-tracking.

V. CONCLUSION

In this paper, we describe an occlusion-aware correlation particle filter target-tracking algorithm based on RGBD data. The proposed algorithm effectively solves some of the problems associated with target occlusion, including the accurate judgment of occlusion and the re-detection of the target. The proposed OACPF algorithm uses a target occlusion judgment mechanism based on a depth image and the associated HOG feature to accurately judge the occurrence of target occlusion. The algorithm also uses maximum likelihood estimation combined with the KCF tracking algorithm to reduce the number of particles required in particle filtering, not only improving the running speed of the algorithm, but also enhancing the accuracy of the particle filter predictions and providing substantial help for target re-detection. Moreover, the adaptive model updating strategy included in the algorithm corrects problems such as model offset, scale variation, and feature loss, and provides the necessary basis for the re-detection of the target. Compared with existing target tracking algorithms, the experimental results show that the proposed OACPF detects target occlusion and tracks the target well. The proposed OACPF method reduces the impact of illumination changes, requires fewer calculations, and achieves better real-time performance and higher accuracy than many existing algorithms.

REFERENCES

[1] S. Chan, X. Zhou, and S. Chen, "Robust adaptive fusion tracking based on complex cells and keypoints," *IEEE Access.*, vol. 5, pp. 20985–21001, 2017.

[2] C. Gong, K. R. Fu, A. Loza, Q. Wu, J. Liu, and J. Yang, "PageRank tracker: From ranking to tracking," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 882–893, Jun. 2014.

[3] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–575, May 2003.

[4] T. Vojir, J. Noskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," in *Proc. SCIA*, Halmstad, Sweden, Jun. 2013, pp. 652–663.

[5] R. P. Tripathi, S. Ghosh, and J. O. Chandle, "Tracking of object using optimal adaptive Kalman filter," in *Proc. ICETECH*, Coimbatore, India, Mar. 2016, pp. 1128–1131.

[6] S. Denman, V. Chandran, and S. Sridharan, "An adaptive optical flow technique for person tracking systems," *Pattern Recognit. Lett.*, vol. 28, no. 10, pp. 1232–1239, Jul. 2007.

[7] W. Cao, Y. Li, Z. He, G. Cao, and Z. He, "Supplementary virtual keypoints of weight-based correspondences for occluded object tracking," *IEEE Access.*, vol. 6, pp. 9140–9146, 2018.

[8] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[9] S. Hare *et al.*, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[10] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 2544–2550.

[11] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2016.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. ECCV*, Florence, Italy, Oct. 2012, pp. 702–715.

[13] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 5000–5008.

[14] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.

[15] T. Bando, T. Shibata, K. Doya, and S. Ishii, "Switching particle filters for efficient visual tracking," *Robot. Auton. Syst.*, vol. 54, no. 10, pp. 873–884, Oct. 2006.

[16] J. Kwon and K. Mu Lee, "Visual tracking decomposition," in *Proc. CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 1269–1276.

[17] L. Čehovin, M. Kristan, and A. Leonardis, "An adaptive coupled-layer visual model for robust visual tracking," in *Proc. IEEE ICCV*, Nov. 2011, pp. 1363–1370.

[18] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[19] M. Isard and A. Blake, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, Aug. 1998.

[20] F. Liu, T. Zhou, C. Gong, K. Fu, L. Bai, and J. Yang, "Inverse nonnegative local coordinate factorization for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1752–1764, Aug. 2017.

[21] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah, "Sequential Monte Carlo tracking by fusing multiple cues in video sequences," *Image Vis. Comput.*, vol. 25, no. 8, pp. 1217–1227, 2007.

[22] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003.

[23] P. Pérez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *Proc. IEEE*, vol. 92, no. 3, pp. 495–513, Mar. 2004.

[24] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[25] X. Mei, H. Ling, Y. Wu, E. P. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling L1 tracker with occlusion detection," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2661–2675, Jul. 2013.

[26] S. Kwak, W. Nam, B. Han, and J. H. Han, "Learning occlusion with likelihoods for visual tracking," in *Proc. ICCV*, Barcelona, Spain, Nov. 2011, pp. 1551–1558.

[27] K. Meshgi, S.-I. Maeda, S. Oba, H. Skibbe, Y.-Z. Li, and S. Ishii, "An occlusion-aware particle filter tracker to handle complex and persistent occlusions," *Comput. Vis. Image Understand.*, vol. 150, pp. 81–94, Sep. 2016.

[28] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[29] W. Yang, M. Zhao, Y. Huang, and Y. Zheng, "Adaptive online learning based robust visual tracking," *IEEE Access.*, vol. 6, pp. 14790–14798, 2018.

[30] F. Liu, C. Gong, X. Huang, T. Zhou, J. Yang, and D. Tao, "Robust visual tracking revisited: From correlation filter to template matching," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2777–2790, Jun. 2018.

[31] S. Song and J. Xiao, "Tracking revisited using RGBD camera: Unified benchmark and baselines," in *Proc. ICCV*, Washington, DC, USA, Dec. 2013, pp. 233–240.

[32] C. K. I. Williams, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. vol. 98, no. 462. Alexandria, VA, USA: American Statistical Association, 2002, p. 489.

[33] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. ICCV*, Venice, Italy, Oct. 2017, pp. 1144–1152.

[34] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *Proc. CVPR*, Las Vegas, Nev, USA, Jun. 2016, pp. 4266–4274.

[35] Y. Li and J. K. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. ECCV*, Zürich, Switzerland, Sep. 2014, pp. 254–265.

[36] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. CVPR*, Las Vegas, Nev, USA, Jun. 2016, pp. 1401–1409.

[37] S. Hannuna *et al.*, "DS-KCF: A real-time tracker for RGB-D data," *J. Real-Time Image Process.*, pp. 1–20, Nov, 2016.

[38] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc ECCV*, Florence, Italy, Oct. 2012, pp. 864–877.

[39] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online Multiple Instance Learning," in *Proc. CVPR*, Miami, FL, USA, Jun. 2009, pp. 983–990.

[40] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. ECCV*, Marseille, France, Oct. 2008, pp. 234–247.

[41] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

YAYU ZHAI received the B.S. degree in bioengineering and master's degree in signal and information processing from the North University of China, Taiyuan, China, in 2015. He is currently pursuing the Ph.D. degree in mechatronic engineering with the Beijing Institute of Technology, Beijing, China.

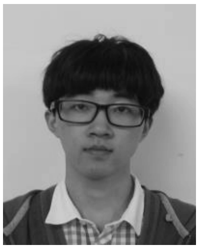
His research interests include signal and information processing, imaging, object detection, object recognition, machine learning, and machine vision.

PING SONG (M'08) received the B.S. degree in automation engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1995, and the M.S. and Ph.D. degrees in mechatronic engineering from the Beijing Institute of Technology, Beijing, China, in 1999 and 2002, respectively. She is currently a Professor of mechanical engineering with the Beijing Institute of Technology.

Her research interests include advanced sensing techniques, fault diagnosis, prognostic and health management, and signal processing.

ZONGLEI MOU received the B.S. and master's degrees in mechanical engineering from the Shandong University of Science and Technology, Shandong, China, in 2009 and 2013, respectively. He is currently pursuing the Ph.D. degree in mechanical engineering with the Beijing Institute of Technology, Beijing, China.

His research interests include advanced sensing techniques, signal processing, and distributed measurement system.



XIAOXIAO CHEN received the B.S. degree in engineering mechanics from the Beijing Institute of Technology, Beijing, China, in 2016, where he is currently pursuing the master's degree in mechanical engineering.

His research interests include embedded system machine learning and computer vision.

XIONGJUN LIU received the B.S. degree in mechatronic engineering from the North University of China, Taiyuan, China, in 2013. He is currently pursuing the Ph.D. degree in mechatronic engineering with the Beijing Institute of Technology, Beijing, China.

His research interests include machine learning, intelligent fault diagnostics, and prognostics and health management.