IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 20, 2018, accepted July 19, 2018, date of publication July 31, 2018, date of current version August 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2861827

Fast Visual Tracking With Robustifying Kernelized Correlation Filters

QIANBO LIU[1,2], GUOQING HU[1], AND MD MOJAHIDUL ISLAM[1]
[1]School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510641, China
[2]Guangzhou Communications Technician Institute, Guangzhou 510540, China

Corresponding author: Guoqing Hu (gqhu@scut.edu.cn)

This work was supported by the National High Technology Research and Development Program of China through the 863 Project under Grant number 2014AA042001.

ABSTRACT Robust visual tracking is a challenging work because the target object suffers appearance variations over time. Tracking algorithms based on correlation filter have presently attracted much attention because of their high efficiency and computation speed. However, these algorithms can easily drift for the noisy updates. Moreover, they are out of action and cannot re-track when trackers failure caused by heavy occlusion or target being out of view. In this paper, we propose a robust correlation filter that is constructed by considering all the extracted target appearances from the initial image to the current image. The numerator and denominator of the filter model are updated separately instead of linearly interpolated only by storing the current model. Strategies, such as reducing feature dimensionality and interpolating correlation scores, are investigated to reduce computational cost for fast tracking. Occlusion and fast motion problems can be effectively solved by the expansion of the search area. In addition, model updates occur under the condition of a confidence metric (i.e., peak-to-sidelobe ratio) threshold. Comprehensive experiments were conducted on object tracking data sets and the results showed that our method performs well compared to the other competitive methods. Moreover, it runs on a single central processing unit at a speed of 69.5 frames per second, which is suitable for real-time application.

INDEX TERMS Adaptive model updating, feature dimensionality reduction, kernel correlation filters, visual object tracking.

I. INTRODUCTION

Visual object tracking is the basic research field in computer vision because of its numerous applications, such as in intelligent vehicles, human-computer interaction, and surveillance [1]. In general, only given the location of a target in the initial image, the task is to estimate the unknown target object situation such as position and scale throughout a video sequence. Despite significant improvements have made during last decade, it's remains a challenging problem because of two reasons. First, constructing a robust and discriminative appearance model during the initial stage of tracking is difficult because object representation is obtained from the initial image without any prior information. Moreover, the propagated and accumulated errors during the tracking process will lead to poor performance in long-term tracking. Second, target appearance will change because of deformation, sudden motion, illumination change, heavy occlusion, and target disappearance in the camera view.

Remarkable progress in visual tracking has been achieved in recent years, which is reflected in various papers of improvement performance on the topic and the existing challenges on multiple performance evaluation benchmarks [2]–[7]. Semi-supervised discriminative tracking approaches [8]–[12] are known for their advantages among tracking algorithms. In particular, correlation filter (CF)-based trackers [13]–[20] have attracted considerable attention due to their excellent performance. The advantage of CF-based trackers can be attributed to the following important characteristics. First, CF-based tracking algorithms perform all operation in Fourier domain which highly increase the computational speed. Furthermore, tracking accuracy improves by using the marginal improvement to decrease the noisy Fourier representation. Second, CF-based trackers use numerous synthetic implicit samples for model training and achieve fast training and detection by applying the circulant structure and convolution theory [17], [19]. Third, Training process of CFs is considered to be a ridge regression problem, where the regression labels generated by a Gaussian function are assigned to the circularly shifted samples of the input image patches. Contrary to the hard labels, the Gaussian

labels are assigned with continuous values ranging from zero to one. Although these algorithms have demonstrated impressive performance on several benchmarks, several underlying issues that severely hamper tracking performance exist. First, these methods update their learned models by moving average schemes with high learning rate to handle appearance changes over time. The update scheme leads to a drift during tracking due to noisy updates. Moreover, these algorithms cannot recover from tracking failures. Second, these trackers cannot successfully handle large-scale variations. Existing scale estimation approaches operate by constructing a 3D CF for jointly estimating translation and scale [16], which is computationally demanding and unsuitable for real-world tracking applications. Third, these trackers may frequently experience the problems of fast motion and occlusions during tracking in an inherently limited search space, which leads to a dilemma in expanding the search area and reducing computational cost for robust tracking.

In the current study, we address the aforementioned problems in terms of several aspects. First, we construct a tracking filter by directly considering all of the previous frames when computing the current model. Then, a dimensionality reduction strategy is used to maintain efficiency, which is also the scheme adopted for scale estimation, as the fast discriminative scale space tracking (fDSST)-based scale variation estimation [21]. To solve the fast motion and occlusion problems, the search area is expanded and the tracker model is updated when the peak-to-sidelobe ratio (PSR) [13], [22] reaches a certain threshold. The main contributions of this study are summarized as follows:

1) We constructed a cost function with a weighted average quadratic error over the frames which comprises all the previous frames and the current frame. The tracker filter that minimized the cost function was updated using all samples from all the previous frames and the current frame by only storing the current learned models.
2) We used strategies, such as feature dimensionality reduction, correlation score interpolation, and the Gaussian radial basis function (RBF) kernel, to reduce computational cost even under an expanded search area. The performance of the tracker remained robust in real time by applying the aforementioned strategies.
3) We comprehensively discussed and compared the proposed algorithm with the concurrent work. Extensive experiments were executed to validate the performances on the online tracking benchmark OTB2013 [6] and OTB2015 [5] datasets. The experimental results showed the efficiency and robustness of our proposed tracker compared with the other competitive methods.

The rest of the paper is organized as follows: In section II, we introduce different trackers that are relevant to our study. In section III, we present our fast tracker with robustifying kernel correlation filter in a detail description. The results of performed experiments are shown in section IV. Finally, the conclusions are drawn in section V.

II. RELATED WORKS

In this section, the CFs and CF-based trackers are the methods that we are mainly focus on. For a comprehensive review on these object tracking approaches, the readers can make a reference in [2], [3], [5], and [6].

CFs are applicable in various fields of computer vision tasks, including object detection and recognition. They have become popular in the tracking community only recently after Bolme *et al.* [13] published their minimum output sum of squared error (MOSSE) tracker in 2010. By mapping the image data from the spatial domain into the frequency domain, the authors showed that the discriminative correlation filter can be trained efficiently with only fast Fourier transforms (FFTs) and pointwise operations. On the basis of the aforementioned strategy, MOSSE achieved excellent performance on a recent tracking benchmark [5] at a remarkable processing speed. Since then, motivated by the strategy of this method, various efforts have been exerted to enhance the robustness and accuracy in tracking process. Most improvements of CF trackers fall into three categories: application of improved features, scale adaptation and conceptual improvements in filter learning.

As demonstrated in [23], object feature is an important factor in visual tracking process. An appropriate feature representation can increase the performance of the tracking methods. Henriques *et al.* [19] extended one-dimensional templates into multi-channels features with HOG [17]. Danelljan *et al.* [15] exploited multi-dimensional color attributes for visual tracking. Li and Zhu [20] incorporated complementary features to strengthen the robustness of the tracker. Recently, deep network features [24] learned for object detection is popular in visual tracking as a feature extractor. The representations in deep features [25]–[28], [29] boost the performance in the discriminability and robustness, but the much computational burden is unsuitable for real-time tracking. It is well to be reminded that Danelljan *et al.* [26] proposed dimensionality reducing strategy in deep features to enhance the computational speed for visual tracking.

To handle object scale variation problem in the visual tracking system, different scale search techniques are applied. Li and Zhu [20] made a strategy of multi-resolution translation filter to achieve scale adaption. Danelljan *et al.* [16] constructed a 3D CF for jointly estimating translation and scale. Huang *et al.* [30] integrated a class-agnostic detection method into a CF-based tracker for scale and aspect ratio adaptability. A part-based CF tracker was developed by Liu *et al.* [31] and the authors designed a Bayesian framework for all the part features to scale evaluate. However, the aforementioned approaches are low in efficiency and computationally demanding. Therefore, to achieve fast and exact scale evaluation in visual tracking remains a challenge.

Conceptually, Henriques *et al.* [17], [19] were the contributors who first successfully applied the kernelized formulation as a theoretical extension to the CF tracker. Thereafter, Ma *et al.* [32] presented that the translation and scale estimation of object in visual tracking operate indepen-

dently and the activation of redetection mechanism during tracking drifting enhances efficiency. Ali *et al.* [33] proposed a heuristic approach comprised correlation, Kalman filter and adaptive fast mean shift to support each other for robust tracking. Recently, Danelljan *et al.* [29] employed an implicit interpolation model to train filters in the continuous spatial domain and used Fast Sub-grid Detection to achieve superior results on the object tracking benchmarks. Galoogahi *et al.* [18], [34] addressed the problem of unwanted boundary effects resulting from learning with circular correlation of implicit patches, the authors proposed that a more discriminative filter could learn from real negative examples which were densely extracted from the background. However, this method imposes heavy computational cost and is unfitting for real-time tracking.

III. PROPOSED TRACKER

In this section, we display a full description of the proposed tracker. For better understanding the proposal, we first provide the tracking pipeline of our approach. Then, we describe briefly the kernelize CF tracker which was applied for the strategies both of the extensive robustifying KCF tracker and scale estimation. We then expand the kernelized CF (KCF) tracker with a robust update scheme. In addition, we provide a whole description of the techniques declared in our proposed tracking, including the fast KCF tracker and adaptive model updating.

A. TRACKING PIPELINE

There are three ordinal parts including model training, object detection and model updating in the overall tracking process system (Fig. 1). The feature dimensionality reduction strategy is used in the three parts to reduce computational cost. First, we use the cyclic shift versions cropped from the search area around centres to train a 2D translation filters and the multi-scale sampling centered at the position to train a 1D scale filter separately (as described in Sections III-C and III-E, respectively). In the object detection process, the location translation filter does correlation with candidate patches in each new frame to find the most relevant location, then multi-scale samplings centered the new position are cropped and scaled filter is applied to find the optimal scale in these scale samples. Finally, a simple but useful model updating strategy of our approach is presented in Section III-F.

B. KCF TRACKER

We give a briefly introduction of the KCF tracking method which is utilized for the strategies both of the extensive

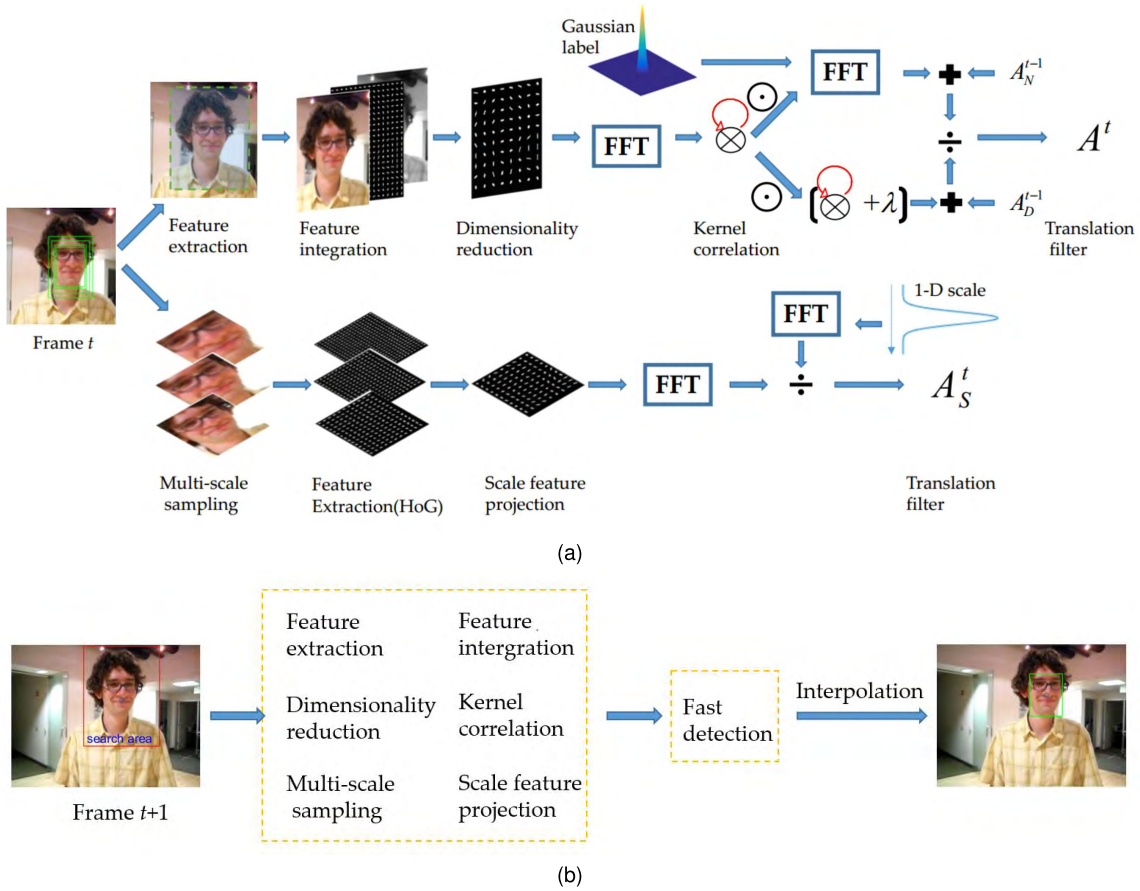


FIGURE 1. Illustrations of our proposed method. The object tracking problem is decomposed into two sections: translation estimation and scale estimation. (a) Model learning and updating at the t-th frame. (b) Target tracking at the $(t+1)$-th frame.

robustifying KCF tracker and scale estimation. The additional details can be found in [17]. During the training of a linear correlation filter, an image patch x of M×N pixels is first cropped. Then the Circulant Matrix is applied to this cropped image data to generate a quantity of implicit training samples $x_{m,n}(m,n) \in \{0,\cdots,M-1\} \times \{0,\cdots,N-1\}$. The regression value y can be generated using the Gaussian function. Gaussian labeling significantly differs from binary labeling. Gaussian labeling takes a value of 1 at the target center and decays rapidly to 0 for other cyclic shift samples, i.e., the score $y(m,n)$ is the label of sample $x_{m,n}$. To find a function $f(z) = w^{\mathrm{T}}z$ for detection at image z correctly, the objective function is constructed for minimizing the squared error over samples $x_{m,n}$. A learning correlation filter w is trained by solving the objective function of a ridge regression problem:

$$w = \min_{w} \sum_{M,N} \left|\left\langle \phi(x_{m,n}), w\right\rangle - y(m,n)\right|^2 + \lambda \left\|w\right\|^2 \tag{1}$$

where ϕ represents the mapping from the original space to the Hilbert space with kernel trick. The dot-product between x and x' in Hilbert space can be calculated as $\left\langle \phi(x), \phi(x')\right\rangle = k(x,x')$, k denotes the kernel function (e.g., Gaussian), and λ denotes the regularization parameter for controlling overfitting. After mapping the inputs of a linear problem into a nonlinear feature space $\phi(x)$, the solution can be defined as a linear combination of training samples: $w = \sum_{m,n} \alpha(m,n)\phi(x_{m,n})$. The solution for the dual form is presented as

$$\alpha = (K + \lambda I)^{-1} y \tag{2}$$

where K is the kernel matrix. The kernel used in KCF is permutation invariant, therefore, the matrix K is circulant. It is possible to diagonalize the matrix K in Fourier domain and the coefficient α can be calculated efficiently in the linear case.

$$A = F(\alpha) = \frac{F(y)}{F(k^{xx}) + \lambda} \tag{3}$$

where F means the Fourier transform. k^{xx} denotes the kernel correlation of x with itself. RBF kernels are defined as $k(x,x') = h(\left\|x - x'\right\|^2)$. Gaussian kernel for a useful special case exhibits its form $k(x,x') = \exp(-\frac{1}{\sigma^2}(\left\|x - x'\right\|^2))$, we obtain

$$k^{xx'} = k(x,x') = \exp\Big(-\frac{1}{\sigma^2}\big(\left\|x\right\|^2 + \left\|x'\right\|^2 - 2F^{-1}(\sum_{c} \hat{x}_c^* \odot \hat{x}_c')\big)\Big) \tag{4}$$

where $\hat{x} = F(x)$, $\hat{x}_c$ denotes the DFT of extracted c-th channel features, $\hat{x}_c^*$ is the complex conjugate of $\hat{x}_c$, and $\odot$ denotes the element-wise product. The Gaussian kernel is used for image data with C channel features. Notably, vector α includes full of the $\alpha(m,n)$ coefficients. The object appearance $\hat{x}$ is updated over time. In the KCF tracker, the learned target appearance $\hat{x}$ and the transformed classifier coefficient A are the two models to be updated.

In detection process, a patch z with the same size x is extracted at the old center in the new frame, and the response scores are calculated as

$$y = F^{-1}(F(k^{zx}) \odot A) \tag{5}$$

where $k^{zx} = k(z,x)$ denotes the kernel correlation of x and z, as defined in (4). Finally, the new position of the object in this frame is located by finding the translation with the maximum value in the response map y.

C. ROBUSTIFYING THE CLASSIFIER BASED ON THE KCF TRACKER

Our proposal method is based on the KCF tracker for the excellent performance in term of its high speed and efficient. The appearance variant of object in tracking make it necessary to incrementally update the object model over time. In the KCF tracker, The transformed classifier coefficient A and the learned target appearance $\hat{x}$ are the two models to be updated. It is computationally expensive to update the object model by minimizing the output errors from all previous results. Thus, the tracker updates the models only by linear interpolation as follows:

$$A^t = (1-\eta)A^{t-1} + \eta A \tag{6a}$$

$$\hat{x}^t = (1-\eta)\hat{x}^{t-1} + \eta \hat{x} \tag{6b}$$

where t is the frame index and $\eta \in (0,1)$ is the learning rate. The above models which do not adopt simultaneously all the previous frames to update the current model can lead to sub-optimal performance. In contrast with the KCF tracker, the model update strategy in MOSSE use all the previous frames to update the current model. However, only single-dimension feature and linear kernels are applied in this update strategy. In the current study, we employ the model update technique of [13] to kernelized correlation classifiers via multi-channel HOG and color naming features.

To provide a robust update scheme for a learning target model, all the appearances $\{x_i\colon i = 1,\cdots,t\}$ of the target extracted from all the previous frames together with the current frame t are adopted.The cost function is constructed using the weighted average quadratic error between the actual results and desired results over these frames. The simplification of the model training and object detection steps in tracking is possible on the condition of the solution limited with only one set of classifier coefficients α. β denotes the weight that controls the relative importance of different frames that are greater than zero. The total cost function can be expressed as follows:

$$\epsilon = \sum_{i=1}^{t} \beta_i \left(\sum_{m,n} \left|\left\langle \phi(x_{m,n}^i), w^i\right\rangle - y^i(m,n)\right|^2 + \lambda \left\langle w^i, w^i\right\rangle \right) \tag{7}$$

where $w^i = \sum_{m,n} \alpha(m,n)\phi(x_{m,n}^i)$. The classifier obtained by minimizing the cost function is of the form:

$$A^t = \frac{\sum_{i=1}^{t} \beta_i Y^i U_x^i}{\sum_{i=1}^{t} \beta_i U_x^i (U_x^i + \lambda)} \tag{8}$$

we define $U_x^i = F(u_x^i)$ as the kernel output in Fourier domain, where $u_x^i(m,n) = k(x_{m,n}, x^i)$ and k is the kernel cross-correlation. The weights β_i are set by using a learning rate parameter η as the sums of the numerator A_N^t and denominator A_D^t of classifier $A^t = A_N^t/A_D^t$ in (8) can be calculated individually by linear interpolation. Inspired by the model updating in MOSSE tracker, we update the numerator A_N^t and denominator A_D^t of classifier A^t separately. The total model in our proposal method is updated using (9). The target appearance $\hat{x}^t$ is updated in the same manner as that in the KCF tracker.

$$A_N^t = (1-\eta)A_N^{t-1} + \eta Y^t U_x^t \tag{9a}$$
$$A_D^t = (1-\eta)A_D^{t-1} + \eta U_x^t (U_x^t + \lambda) \tag{9b}$$
$$\hat{x}^t = (1-\eta)\hat{x}^{t-1} + \eta\hat{x} \tag{9c}$$

Notably, the models can be updated only by using the current image data instead of all the previous appearances. Simply by storing the current model $\{A_N^t, A_D^t, \hat{x}^t\}$, this scheme makes the models of new frame to be updated using (9). Moreover, it ensures that the computational increase has tiny effect on tracker speed. Similar to the conventional KCF tracker, the learned appearance $\hat{x}^t$ is used in detection step to calculate the response map y for the next frame $t+1$.

D. FAST KCF TRACKER

Several strategies are investigated to increase the computational speed of our proposed tracker. Two approaches for increasing the computational speed required both in the training and the object detection steps of the multi-dimension KCF described in Sections III-B are exploited. These approaches contain: the feature dimensionality reduction which lower the computational cost using principal component analysis (PCA) and the fast sub-grid detection with interpolation method in correlation scores.

1) DIMENSIONALITY REDUCTION

The high-speed in the KCF tracker is attributed to the image data computed in Fourier domain. Identically, the computational speed in our approach is dominated by the FFT. There is a linear relationship between the number of FFT computations and the feature dimensionality, since each feature dimension is required one FFT operation in the tracking steps of training (3) and detection (5). To increase the tracking speed, we have a strategy of dimensionality reduction. This adaptive dimensionality reduction technique based on the standard PCA can preserve useful information and boost computation speed. However, the smooth subspace update scheme similar to that in [15] is not required owing to the simplicity of the Gaussian kernel.

Through the linearity of the Fourier transform, the numerator A_N^t (9a) and denominator A_D^t (9b) of the learned filter $A^t = A_N^t/A_D^t$ can be equivalently obtained from the output of the kernel function $U_x^t = F(u_x^t)$, where $u_x^t(m,n) = k(x_{m,n}^t, x^t)$. To reduce the dimension of the learned target template $\hat{x}^t = (1-\eta)\hat{x}^{t-1} + \eta\hat{x}$, let the learned appearance $\hat{x}^t$ be the D_1-dimension in original space. a $D_1 \times D_2 (D_1 \gg D_2)$ projection matrix P_t with orthonormal column vectors is constructed using the dimensionality reduction technique, where D_2 is the compressed dimensionality of the feature data in subspace. The matrix P_t is applied to generate the new feature map $\hat{x}^t$ with D_2-dimensional representation by linearly mapping $\hat{x}^t = P_t^{\mathrm{T}}\hat{x}^t(m,n), \forall m,n$. We get the matrix P_t by minimizing the reconstruction error of the learned object template.

$$\epsilon_1 = \frac{1}{M \times N}\sum_{m,n}\left\|\hat{x}^t(m,n) - P_t P_t^{\mathrm{T}}\hat{x}^t(m,n)\right\|^2 \tag{10}$$

The indexes m and n range across all elements in the template $\hat{x}^t$ with multi-channel features. The minimization of (10) is under the orthonormality constraint $P_t P_t^{\mathrm{T}} = I$. A solution of the optimal P_t is obtained by calculating the auto-correlation matrix and doing eigenvalue decomposition.:

$$C_t = \frac{1}{M \times N}\sum_{m,n}(\hat{x}^t(m,n) - \bar{x}^t)^{\mathrm{T}}(\hat{x}^t(m,n) - \bar{x}^t) \tag{11}$$

where $\bar{x}^t$ is the mean of $\hat{x}^t$, the D_2 eigenvectors corresponding to the largest eigenvalues of C_t are the columns of P_t.

The compressed training samples $\dot{x}^t = P_t^{\mathrm{T}} x^t$ are applied to update the filter. The numerator A_N^t (9a) and denominator A_D^t (9b) of the learned filter A^t are updated separately using the Fourier transformed kernel output $U_x^t = F(\dot{u}_x^t)$, where $\dot{u}_x^t(m,n) = k(\dot{x}_{m,n}^t, \dot{x}^t)$, instead of u_x^t. In the detection step of tracking, the test sample z is first compressed by the projection matrix P_t. Then similarly to (5) by applying the filter on the compressed target template $\dot{x}^{t-1} = P_{t-1}^{\mathrm{T}}\hat{x}^{t-1}$ and the compressed sample $\dot{z}^t = P_{t-1}^{\mathrm{T}} z^t$, the correlation scores are obtained as follows :

$$y = F^{-1}(U_z \odot A^{t-1}) \tag{12}$$

2) INTERPOLATION CORRELATION SCORES FOR DETECTION

In the training and detection stages, the coarse features of samples have a grid stride greater than one pixel, this increases computation speed for the reason that the size of the operated FFT reduces. Consequently, we can compute the detection score (5) only on a coarse grid. An approach of sub-grid interpolation is employed to calculate the detection score in pixel-dense. This interpolation with trigonometric polynomials is especially suitable as the detection score (5) can be efficiently executed by performing the computed DFT coefficients. Let $\hat{y} = F(y)$ be the DFT of the detection score evaluated on sample z. The interpolated detection scores $y(u,v)$ at the pixel-dense position $(u,v) \in [0,M) \times [0,N)$ in z are obtained using

$$y_t(u,v) = \frac{1}{M \times N}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\hat{y}_t(m,n)e^{i2\pi(\frac{m}{M}u + \frac{n}{N}v)} \tag{13}$$

where i denotes the imaginary unit. To obtain the interpolated scores $\hat{y}_t$, we expand the high frequencies of $\hat{y}$ in (5) by zero-padding to the same size with the size of the interpolation grid. The interpolated scores y_t are then achieved by executing the inverse DFT of the expanded $\hat{y}_t$.

E. SCALE ADAPTATION MECHANISM

In contrast with the 3D CF for incorporating both estimation of scale and translation presented by Danelljan *et al.* [16], we construct a 2D scale feature projection to train a scale regression model using CF.

Let the object size be $W \times H$ in pixels and S denote the number of scales. Then, S image patches are extracted around the estimated position. The size of each image patch is obtained by $\{S_p = a^p W \times a^p H | p \in \{[-\frac{S-1}{2}], \cdots [\frac{S-1}{2}]\}\}$, where a is the scale factor. We uniformly resize all the cropped patches to $W \times H$, and then use the HOG feature descriptor of each sampled patch to map a feature vector. The scale filter training sample $f_{t,scale}$ is constructed on the above vectors. Evidently, the sample $f_{t,scale}$ consists of a d-dimensional feature vector $f_{t,scale}(p) \in R^d$ for each p-th scale patch. Compared to the feature dimensionality $d \approx 1000$ in the 3D correlation scale filter case, the number of training samples $S = 17$ is significantly small. Obviously, the scale sample $f_{t,scale}$ is a compressed S feature dimension without any information loss. The same is true for scale template $u_{t,scale}$, where a simple linear interpolation $u_{t,scale} = (1-\eta)u_{t-1,scale} + \eta f_{t,scale}$ is used.

To train a scale filter of the proposal tracker, we utilize the dimensionality reduction strategy with its properties presented in Section III-D. Two projection matrices, $P^f_{t,scale}$ and $P^u_{t,scale}$, are constructed for efficiency based on $f_{t,scale}$ and $u_{t,scale}$, respectively. Then, the compressed sample and template can be obtained with fully information by using $\dot{f}_{t,scale} = (P^f_{t,scale})^{\mathrm{T}} f_{t,scale}$ and $\dot{u}_{scale} = (P^u_{t,scale})^{\mathrm{T}} u_{t,scale}$. Let $f^p_{t,scale}$ be p-th scale sample with the object feature descriptor, a regression target score $y_p = \exp(-\frac{1}{2\sigma^2}(p - \frac{S}{2})^2)$, is assigned to this scale sample, where $\{y_p\}$ is 1D. We utilize (3) to train the CF scale filter. The response result of (5) is scalar and calculated at a compressed test sample $\dot{z}_{t,scale} = (P^u_{t,scale})^{\mathrm{T}} z_{t,scale}$. To mitigate ambiguity, we define $g(f^p_{t,scale})$ as the response result from (5). The optimal scale p^* of the target can then be deduced by

$$p^* = \underset{p}{\operatorname{argmax}} \{g(f^p_{t,scale}) | p \in S\} \tag{14}$$

F. MODEL UPDATE STRATEGY

To estimate the likelihood that a target can be tracked effectively, a confidence metric namely PSR is adopted. Generally, PSR is employed in signal processing field to measure peak strength in a response map. Motivated by [13] and [22], we generalize PSR to our tracker system as a trackable confidence function for a test frame. We define PSR as follow:

$$\mathrm{PSR}(z) = \frac{\max(y_t) - u_\phi(y_t)}{\sigma_\phi(y_t)} \tag{15}$$

where z denotes an image patch for detection; y_t stands for typically a output response map computed in the test patch; and ϕ denote the sidelobe region centred around the peak, which is set to 15% of the whole response map area in this study. μ_ϕ and σ_ϕ denote the mean value and standard deviation of y_t, excluding the sidelobe area ϕ, respectively. Evidently, the function PSR(z) becomes considerable when the response peak is strong. Therefore, PSR(z) can be considered as the confidence of a sample to determine whether it is tracked properly. During tracking, some parts of the target may be invisible due to occlusion or out of view condition, which makes the tracker result unreliable. Therefore, current appearance should not be used for updating. The current appearance and other models should be useful for model updates if the value of PSR is large. The threshold T of PSR is 16 in the experiments.

IV. EXPERIMENTAL RESULTS AND ANALYSES

Here, our Pcakcf tracker is evaluated and compared with state-of-the-art methods on OTB [5], [6]. First, we provide the parameters and evaluation protocol used in our experiments. Then, we display the quantitative results performed on the OTB2013 [6] benchmark compared with the fDSST tracker because they belong to the same scale estimation method. We also extend a further comparison implemented on this benchmark with the representative trackers. Furthermore the experimental results of the overall performance achieved on OTB2015 [5] are presented to demonstrate the superiority of our Pcakcf tracker compared with the state-of-the-art algorithms. Finally, we execute component analysis to distinguish the contribution of the different strategies in our proposed tracker.

A. EXPERIMENTAL DETAILS AND METHODOLGY

Our proposal tracker is executed in MATLAB. All the experiments are performed using an Intel I5-5200U Core 2.2 GHz CPU with 8 GB RAM. The parameters used in our Pcakcf method are described as follows: Gaussian kernel correlation sigma is 0.2, regulation term is 0.01, interpolation factor is 0.025, compressed feature size D2 is 21, HoG cell size is set to 4× 4, number of HOG orientation bins is set to 9, scale factor a is 1.02, interpolation number of scales S_1 is 33 and padding is 1.8. All the parameters are fixed in our proposed method for all experiments and videos.

In order to get fair and rigorous comparison, our tracker is quantitatively evaluated on OTB datasets [5], [6], which contain 51 and 100 challenging image sequences separately. We use two standard evaluation metrics to evaluate the tracking results on the these datasets. The first one is distance precision (DP) which takes more concern on the distance. The DP score is calculated as the percentage of the correctly tracked frames in a sequence, where the Euclidean distance between the estimated location of the target and the ground truth centroid is smaller than a given threshold. To compare the performance of different trackers, in this study, we provide the precision plot where the DP scores are obtained at

Algorithm 1 The Proposed Tracking Algorithm

Input: Image I_t, previous bounding box $b_{t\hat{a}\acute{L}'1} = (x_{t-1}, y_{t-1}, s_{t-1})$, A_N^{t-1}, A_D^{t-1}, P_{t-1}, $\hat{x}^t$ and A_s^{t-1}.
Output: Estimated bounding box $b_t = (x_t, y_t, s_t)$

repeat
  Crop out the image patch z from I_t according to the center position (x_{t-1}, y_{t-1}) and scale s_{t-1}.
  Extract features and reduce feature dimensions using P_{t-1}.
  // Translation estimation
  Compute A^{t-1} and U_z.
  Calculate their response maps in the Fourier domain using (12).
  Interpolate their response maps in the Fourier domain using (13).
  Estimate the target positon (x_t, y_t) by the maximum value in the final response map.
  Calculate the PSR value using (15) for updating.
  // Scale estimation
  Construct 2D scale feature projection z around (x_t, y_t) and compute $\hat{y}_S^t$ using A_S^{t-1} in the Fourier domain.
  Interpolate their scale response maps $\hat{y}_S^t$ in the Fourier domain to infer s_t.
  // training
  Crop out the image patch x from I_t according to the center position (x_t, y_t) and scale s_t.
  Extract features and compute the new project matrix P_t.
  Reduce feature dimensions using P_t.
  Compute $Y^t U_x^t$, $U_x^t(U_x^t + \lambda)$, and A_S^t.
  // Updating
  if PSR$\geq T$ **then**
    update A_N^t, A_D^t, and $\hat{x}^t$ using (9) and A_S^t using (3).
  end if
until end of image sequence

a location error threshold of 20 pixels to rank the tracking results. The second is overlap precision (OP) which focuses on the scale change of an object, the OP score (success rate) is calculated as the percentage of frames in a video whose overlap rate exceeds a certain threshold. The overlap rate is defined as $O = \text{Area}(r_g \cap r_t)/\text{Area}(r_g \cup r_t)$, where r_t is the estimated bounding box, r_g is the ground truth bounding box, $\cap$ and $\cup$ denote the intersection and union of two regions in pixels, respectively. In the tables, the mean OP is obtained at a threshold of 0.5. The success plot is also provided for the tracking results. The success plot shows OP scores across all sequences with the range of the overlap thresholds vary from 0 to 1. The overall performance for each tracker is ranked by using the area under the curve (AUC) of each success plot. These two different plots are obtained by using the one-pass evaluation(OPE) over all dataset videos. We also compare the tracking speed of different methods in frames per second (FPS) to determine whether a method is suitable or not in real time application.

B. PCAKCF AND FDSST

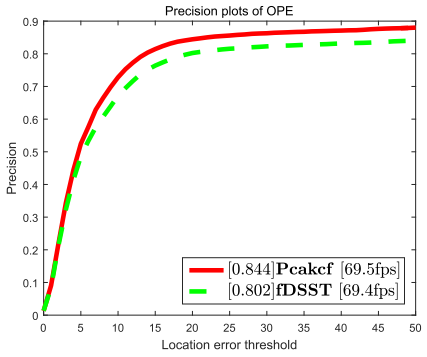
In this section, the experiments are implemented across all 51 videos on the OTB2013 [6]. We compare our robust tracker (Pcakcf) presented in Section III-C with the fDSST tracker for the same scale variation estimation. Moreover, the dimensionality reduction technique is utilized in both trackers for increasing track speed. Our method, furthermore, applies a PSR threshold to improve the robustness of the update strategy. These two approaches are compared in the mean OP (%) and DP (%). Table 1 and Fig. 2 show that our approach acquires a gain of 3.6% and 4.1% in mean OP and DP, respectively. Including the improvement in performance, our method operates at nearly the same speed with the compared fDSST tracker.

C. EXPERIMENTAL RESULTS ON THE OTB2013 DATASET

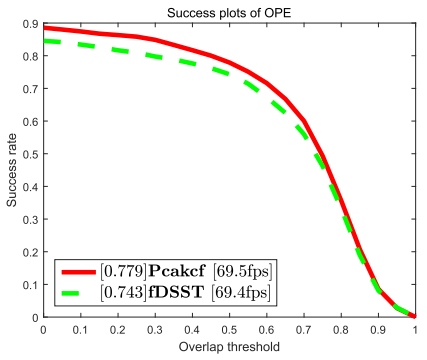
The comparison on the OTB-2013 dataset between our proposed tracker and other 42 trackers is reported in Fig. 3. These

TABLE 1. Comparison of Pcakcf and the fDSST tracker. The mean OP (%) and DP (%) across all 51 sequences on the OTB dataset are displayed. The superior results are presented in red. Our method improves performance significantly while operating at nearly the same mean FPS with the fDSST tracker.

	Mean OP	Mean DP	Mean FPS
fDSST	74.3	80.2	69.4
Pcakcf	77.9	84.4	69.5



(a)



(b)

FIGURE 2. Quantitative evaluation of these two methods on the OTB2013 dataset. Distance precision and overlap success plots using OPE. (a) precision plot with the DP scores at 20 pixels, whereas (b) success plot with OP scores at a threshold of 0.5 intersection over union (IoU).

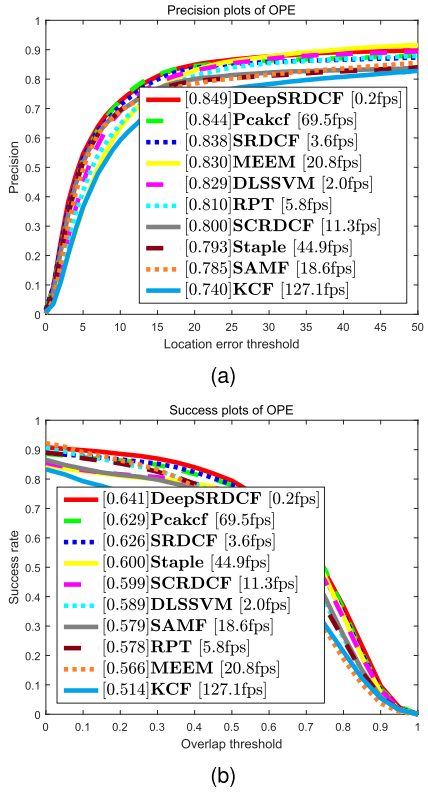


FIGURE 3. Quantitative evaluation of these two methods on the OTB2013 dataset. Distance precision and overlap success plots using OPE. (a) precision plot with the DP scores at 20 pixels, whereas (b) success plot with AUC scores.

state-of-the-art trackers comprise three categories: 1) the 29 conventional trackers provided in [6]; 2) some classic tracking-by-learning approaches, such as DLSSVM [35], MEEM [10], Struck [11] and TGPR [36]; 3) the recent CF based trackers, including KCF [17], DSST [16], RPT [22], SAMF [20], SRDCF [14], Staple [37], SCRDCF [38], DeepSRDCF [14] and MOSSE. We only showed the experiment results of the top ten trackers with one-pass evaluation (OPE). Although the best performance of the tracker is DeepSRDCF which obtains an average precision score of 84.9% and a success score of 64.1%, the slowly speed at the 0.2 fps is unfit for real-time application. Our proposed tracker takes the second place with a small inferior, performing the average precision score of 84.4% and the average success score of 62.9% respectively. Notably, our tracker runs at the 69.5 fps speed which is more than 300 times that of the DeepSRDCF tracker. Compared to the KCF tracker, our proposal tracker achieved a great gain of 10.4% and 11.5% in the average precision rate and the success rate respectively.

D. EXPERIMENTAL RESULTS ON THE OTB2015 DATASET

We compared the proposal approach with other nine representative trackers: SAMF [20], DSST [16], KCF [17], TGPR [36], fDSST [21], MUSTer [39], LCT [32], CSK [19] and STC [40]. The experiments are extended on OTB2015 with 100 sequences which is more challenging. To achieve quantitative results for the compared methods, we employ three evaluation criterions, namely, one-pass evaluation (OPE), temporal robustness evaluation (TRE), and spatial robustness evaluation (SRE), which are shown in Fig. 4. We follow the protocol and display the DP scores at a threshold of 20 pixels and OP scores at a threshold of 0.5 overlap rate. As shown in Table 2, our proposal approach performs favorably against the compared methods in terms of the mean distance precision and the mean overlap precision.

Table 2 presents the experimental results of compared methods. Compared to the LCT and MUSTer tracker, our proposal tracker obtains a lightly better performance both in mean DP and mean OP with a much higher tracking speed. It also shows that the KCF tracker achieves a mean DP of 69.6% and a mean of OP of 55.1% with the sixth place both in precision rate and success rate, for this method employs a kernelized CF for translation estimation, The SAMF tracker obtains a mean OP of 67.4% and a mean DP of 75.1%, the significant gain of 12.3% in success rate compared with the KCF tracker due to a multi-resolution filter approach for scale estimation in SAMF tracker. The DSST tracker, which is the first to construct a separate 1D scale correlation filter for scale estimation, obtains a mean DP of 69.5% and a mean of OP of 53.7%. The fDSST tracker, which improves the computational speed of the DSST method using the feature dimensionality reduction technique, obtains a mean DP of 72.2% and a mean of OP of 66.2%. In contrast with the SAMF tracker and the fDSST tracker, our Pcakcf approach is based on learning a robust correlation filter by training the samples from all previous frames, while exploiting a simple but efficiency scale filter for scale estimation that is similar to fDSST method. Our tracker outperforms fDSST by 5.5% and SAMF by 2.6% in terms of mean DP. Similarly, the highest mean OP score of our tracker shows the superiority over the existing trackers. Notably, our proposal operates in real time (68 in mean FPS) including the superior performance both in precision rate and success rate.

Fig. 4(d) displays the success plot of OPE to the trackers for comparison. In this legend, the success rate is reported using AUC score for each tracker over all the 100 sequences in the dataset, all the trackers in this comparison are ranked to show

TABLE 2. State-of-the-art comparison (Best: red.).

	Pcakcf	MUSTer	LCT	SAMF	fDSST	KCF	DSST	TGPR	CSK	STC
Mean DP	0.777	0.774	0.762	0.751	0.722	0.696	0.695	0.643	0.518	0.507
Mean OP	0.704	0.683	0.701	0.674	0.662	0.551	0.537	0.535	0.411	0.314
FPS	68	10.1	16.3	18.2	65.9	125	24.4	0.7	188.3	264.6

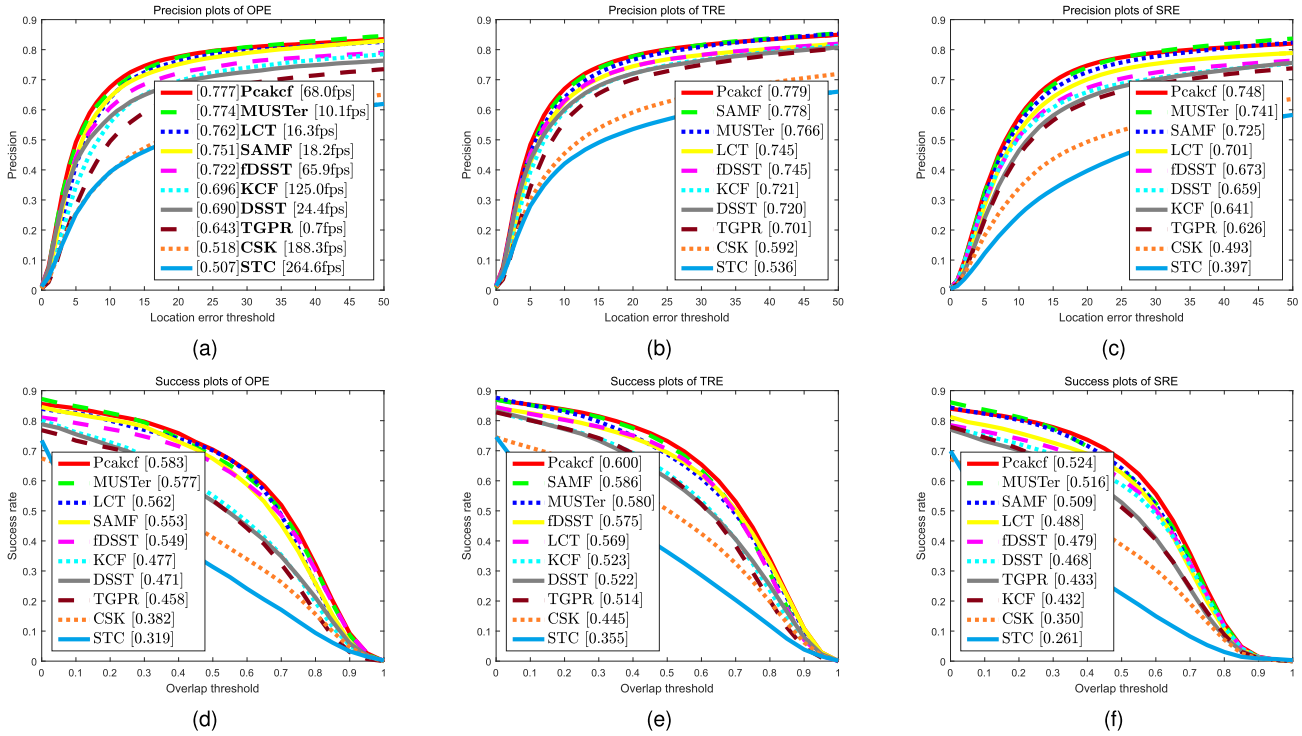


FIGURE 4. The experiments on the OTB2015 dataset. Quantitative evaluation of success and precision plots using OPE, TRE, and SRE. the precision plots with the distance precision scores at 20 pixels, (a), (b) and (c), whereas the success plots contain the overlap success scores with AUC, (d), (e) and (f).

their performances. Obviously, our approach takes the first place with an AUC score of 58.3% which outperforms the other nine trackers. It is noteworthy that our method obtains a gain of 10.6% and 3.4% in AUC score compared with the KCF and the fDSST track separately.

E. ATTRIBUTE-BASED EVALUATION

The performance of a tracker can be affected by serval factors in visual tracking system. To describe the various challenges, 11 different attributes, namely: scale variation, illumination variation, out-of-plane rotation, occlusion, background clutter, deformation, motion blur, fast motion, in-plane rotation, out of view and low resolution, are adopted to annotate the sequences on the OTB dataset. Using these attribute, we evaluate the performance of trackers in different aspects. In Tables 3 and 4, the attribute-based evaluation results are displayed in terms of distance precision and overlap success on OTB2015.

Table 3 lists the DP scores at a threshold of 20 pixels for the 11 attributes. For the better understanding, we show the results of only top 10 methods on the OTB 2015 dataset. Our approach presents superior results in 8 out of 11 attributes and takes second place on the attributes of illumination

TABLE 3. DP scores at a threshold of 20 pixels over all attributes on the OTB2015 dataset. Best results are displayed in red, second best: blue (for our approach only).

	Pcakcf	MUSTer	LCT	SAMF	fDSST	KCF	DSST	TGPR	CSK	STC
Scale variation (64)	0.733	0.710	0.681	0.705	0.662	0.633	0.662	0.599	0.448	0.449
Illumination variation (38)	0.779	0.782	0.746	0.715	0.746	0.719	0.723	0.633	0.482	0.549
Out-of-plane rotation (63)	0.763	0.744	0.746	0.739	0.666	0.677	0.670	0.642	0.489	0.472
Occlusion (49)	0.763	0.734	0.682	0.726	0.640	0.630	0.615	0.594	0.428	0.434
Background clutter (31)	0.809	0.784	0.734	0.689	0.779	0.713	0.702	0.593	0.574	0.556
Deformation (44)	0.713	0.689	0.689	0.686	0.611	0.617	0.568	0.630	0.451	0.440
Motion blur (29)	0.719	0.678	0.669	0.655	0.680	0.601	0.611	0.529	0.355	0.355
Fast motion (39)	0.716	0.683	0.681	0.654	0.690	0.621	0.584	0.533	0.397	0.333
In-plane rotation (51)	0.781	0.773	0.782	0.721	0.727	0.701	0.724	0.659	0.514	0.477
Out of view (14)	0.688	0.591	0.592	0.628	0.578	0.501	0.487	0.493	0.350	0.276
Low resolution (9)	0.750	0.747	0.699	0.766	0.675	0.671	0.708	0.622	0.445	0.489

TABLE 4. Overlap success scores of AUC over all attributes on the OTB2015 dataset. Best results are displayed in red, second best: blue (for our approach only).

	Pcakcf	MUSTer	LCT	SAMF	fDSST	KCF	DSST	TGPR	CSK	STC
Scale variation (64)	0.537	0.512	0.488	0.495	0.497	0.394	0.409	0.404	0.318	0.284
Illumination variation (38)	0.585	0.600	0.566	0.534	0.563	0.479	0.489	0.452	0.368	0.345
Out-of-plane rotation (63)	0.548	0.537	0.538	0.536	0.499	0.453	0.448	0.455	0.354	0.308
Occlusion (49)	0.572	0.554	0.507	0.540	0.484	0.443	0.426	0.429	0.331	0.288
Background clutter (31)	0.595	0.581	0.550	0.525	0.585	0.498	0.477	0.428	0.410	0.369
Deformation (44)	0.538	0.524	0.499	0.509	0.469	0.436	0.412	0.455	0.337	0.293
Motion blur (29)	0.561	0.544	0.533	0.525	0.536	0.459	0.467	0.429	0.308	0.226
Fast motion (39)	0.553	0.533	0.534	0.507	0.547	0.459	0.442	0.420	0.329	0.218
In-plane rotation (51)	0.553	0.551	0.557	0.519	0.541	0.469	0.485	0.462	0.379	0.312
Out of view (14)	0.530	0.469	0.452	0.480	0.457	0.393	0.374	0.373	0.250	0.247
Low resolution (9)	0.464	0.415	0.399	0.425	0.429	0.290	0.314	0.344	0.234	0.232

variation, in-plane rotation and low resolution with scores of 0.779, 0.781 and 0.750, respectively, which are slightly lower than the top-ranking algorithm. Table 4 shows the AUC scores for the 11 attributes. Furthermore, our approach outperforms existing trackers on the 9 attributes. In the sequences annotated with attributes, namely, scale variation, out-of-plane rotation, occlusion, and out of view, our method outperforms the adaptive feature dimensionality reduction technique based on the fDSST tracker in terms of distance precision scores at a threshold of 20 pixels by 7.1%, 9.7%,12.3%, and 11%, respectively, and in AUC scores by 4.0%, 4.9%, 8.8%, and 7.3%, separately. These results show that our tracker achieves superior performance in the scenarios with occlusion and out of view while accurately estimates object with scale variation.

F. COMPARISON OF ROBUSTNESS TO INITIALIZATION

Visual tracking can be sensitive to initialization. To evaluate robustness to initialization, we followed the protocol proposed in [6]. Two different initialization criteria, namely, TRE and SRE, were employed. For TRE performance, each sequence is partitioned into 20 segments, then tracker is evaluated by initializing at different frames with the ground truth bounding box. In the case of SRE, SRE is performed by adding some slight perturbation to the ground truth bounding box in the first frame. A tracker is calculated on each sequence with 12 different initializations including four scale shifts and eight spatial shifts. We refer to [6] for additional details.

The experimental results of the robustness evaluation are presented in Fig. 4. In the precision plots for TRE and SRE, our approach performs favorably compared with the existing approach, with scores of 0.779 and 0.748, respectively. Similarly, in the success plots for TRE and SRE, our tracker takes the top place, with scores of 0.600 and 0.524, respectively, which are better than those of the SAMF and MUSTer tracker. Notably, for increasing computational speed, the fDSST tracker employs a strategy to reduce feature dimensionality using PCA. In summary, our approach performs excellent with a consistent gain compared with the fDSST tracker in the four evaluations.

G. QUALITATIVE ANALYSIS

We compared our proposed method with other four state-of-the-art trackers: SAMF [20], fDSST [21], KCF [17] and TGPR [36]. The ten sequences in Fig. 5 used in this study include various challenges, such as background clutter, illumination variation, occlusion, motion blure and scale variation. Moreover, the center location errors (CLEs) obtained by the five trackers on the ten videos are presented in Fig. 6. CLE is defined as the Euclidean distance between the ground truth and estimated centers.

In the sequences *couple* and *soccer*, the main challenge is to handle background clutter with target scale variation. Among the existing trackers, only Pcakcf tracks the target in both videos with a low location error and a high overlap ratio. The compared trackers are prone to drifting and exhibit a high CLE. These trackers also fail to handle the occlusions and significant clutter in these two sequences. In these two sequences our tracker not only track the target properly but also handle the scale changes accurately.

In the *singer1* and *shaking* sequences, most of approaches fails to detect the target due to the indoor lighting condition and scale variation. Again, our approach obtains satisfactory performance with a low CLE and demonstrates robustness in these scenarios. Furthermore, our tracker is able to keep tracking the target correctly throughout the sequences with an accurate scale variation estimate.

In the *jogging-1* and *tiger2* sequences, the compared trackers struggle due to the heavy occlusion condition and target deformation, whereas our tracker can robustly handle these factors with a relatively low CLE. Although our approach and SAMF nearly have the same result in the *jogging-1* sequence, our tracker is more robust than the SAMF tracker in the *tiger2* video. The good performance is owed to the application of an expanded search area in our tracker.

In the sequences *dog1* and *liquor*, most of the compared trackers are capable of estimating scale variations.

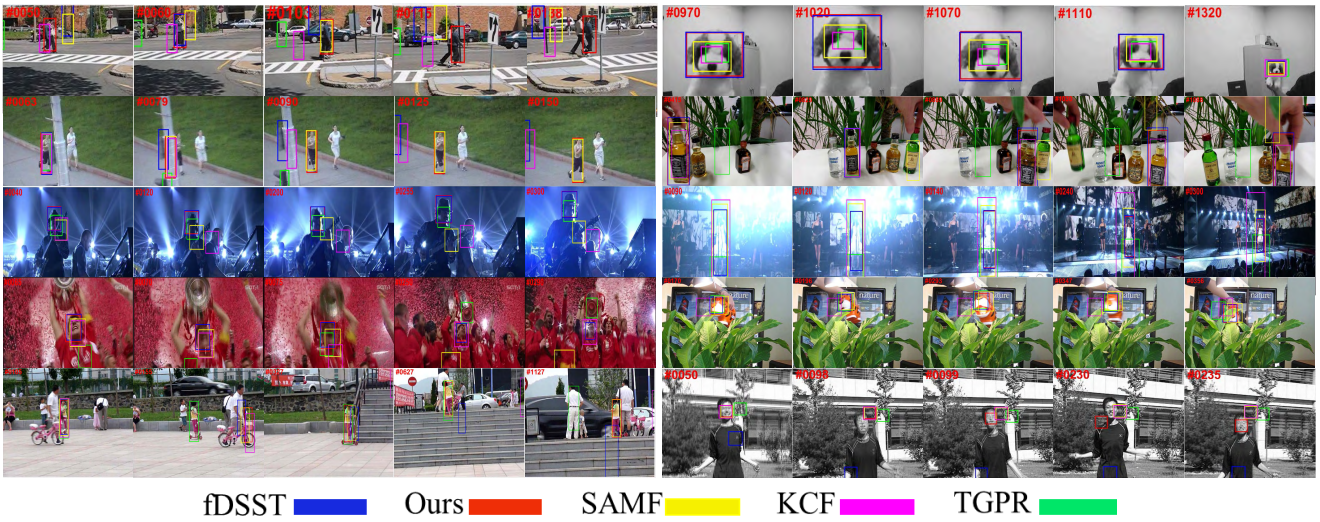


FIGURE 5. Tracking of bounding boxes produced by our proposed tracker and the fDSST [21], SAMF [20], KCF [17], and TGPR [36] trackers on several key frames of ten challenging sequences (from left to right,from top to bottom: *couple, dog1, jogging-1,liquor, shaking, singer1, soccer, tiger2, girl2 and jumping*).

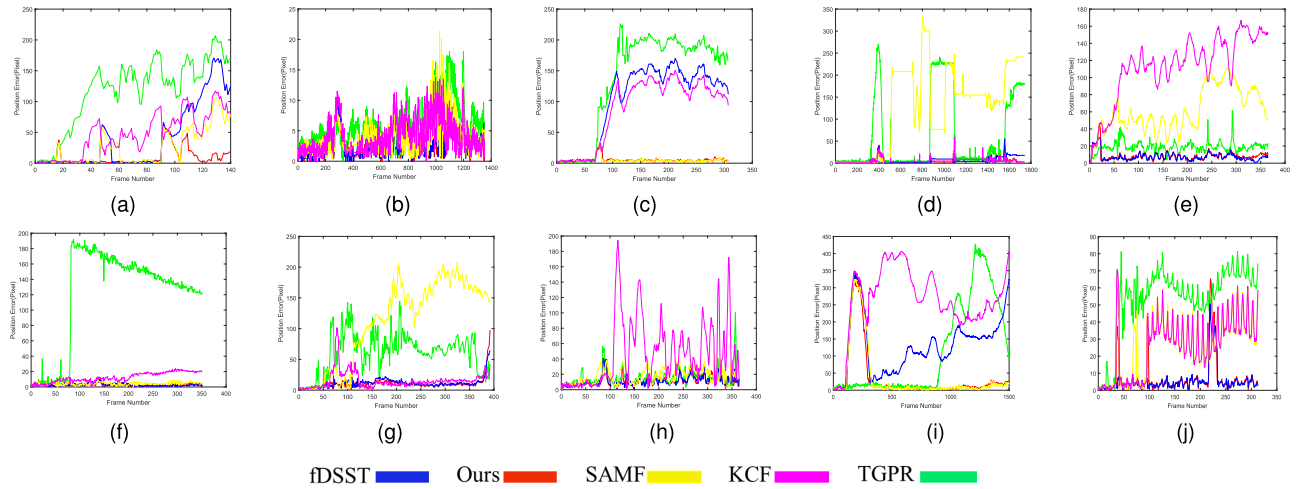


FIGURE 6. Comparison of CLE results achieved by our proposed tracker and four state-of-the-art trackers on ten challenging sequences shown in Fig. 5. (a) couple. (b) dog1. (c) jogging-1. (d) liquor. (e) shaking. (f) singer1. (g) soccer. (h) tiger2. (i) girl2. (j) jumping.

However, the high CLE shows that these trackers suffer from a significant scale and translation drift in the presence of rotating motions and fast scale changes. Both our tracker and fDSST tracker can accurately estimate the target translation and scale despite the aforementioned factors.

In the sequences *girl2* and *jumping*, the motion blur is the main challenge for these trackers. The CLE shows that these compared trackers are prone to drift and fail to maintain long-term tracking. However, our proposal tracker can recover from tracking failures in both sequences. Furthermore our tracker can cope with the target deformation in the *girl2* videos and the fast motion in the *jumping* videos respectively during the correctly tracking.

H. COMPONET ANALYSIS

To demonstrate the contribution of the different component used in our approach, we also execute component analysis. First, we performed four trackers based on the our robust KCF tracker with scale estimation strategy by individually integrating multiple features: the tracker with a searching padding which is set to 1 as the padding size of KCF (namely, RkcfScaleS1), multiple features with the a expand search area where the padding is set to 1.8 (namely, RkcfScale); low dimensional features using PCA strategy (namely, PcaRkcfNopsr) and our proposal tracker with model updating strategy (namely, Pcakcf). The comparative results are presented in Fig. 7. The comparisons show that, compared to the KCF tracker, the performance of RkcfScaleS1 integrated multiple features has only a slight decrease in mean DP scores but a gain of 3.9% in success rate scores because of the limited training samples generated by the small searching area. With the expansion of searching area, the performance of RkcfScale tracker has improvement in precision rate and success rate by 6.2% and 9.0% respectively. Thanks to the

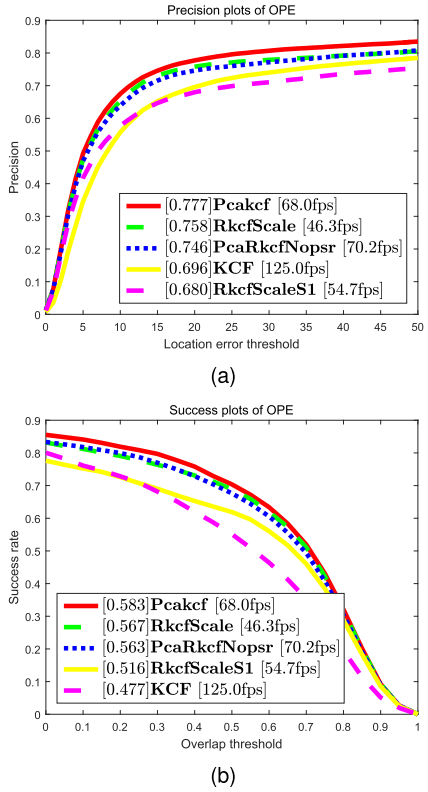


FIGURE 7. The comparisons are performed between the robust KCF trackers using the different components, the experiments are implemented on the OTB-2015 dataset. (a) Precision plots of OPE. (b) Success plots of OPE.

dimensionality reducing strategy, the PcaRkcfNopsr tracker has a gain of 51.6% in tracking speed with a slight decrease in performance. The contribution of the model updating strategy applied in the Pcakcf tracker is the improvement 3.1% in precision rate and 2.0% in success rate respectively. Then we draw the conclusions that the major contribution to the improvement of our proposal performance is the robustifying classifier with the expansion of searching area. The fast speed of our tracker comes from the feature dimension reducing strategy. Our model updating method can also effective in the tracking system.

V. CONCLUSIONS

In this study, we proposed a robust classifier constructed by considering all the extracted appearances of a target from the first frame to the current frame, which was updated only by storing the current model. Furthermore, we investigated strategies to reduce the computational cost of our tracking approach, which allowed us to use a large target search space for learning and detection without sacrificing real-time performance. Finally, the qualitative and qualitative results clearly demonstrated that our approach provided improvement over the fDSST tracker and other representative trackers. Extensive experiments results showed that our method exhibited promising performance in terms of accuracy and robustness. The component analysis presented the effectiveness of the proposed strategies.

REFERENCES

[1] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3D traffic scene understanding from movable platforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 1012–1025, May 2014.

[2] M. Kristan *et al.*, "The visual object tracking VOT2013 challenge results," in *Proc. ICCV Workshops*, Dec. 2013, pp. 98–111.

[3] M. Kristan *et al.*, "The visual object tracking VOT2014 challenge results," in *Proc. ECCV Workshops*, vol. 8926, 2015, pp. 191–217.

[4] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 564–586.

[5] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[6] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[7] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. ECCV*, vol. 9905, 2016, pp. 445–461.

[8] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[9] 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.

[10] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.* Switzerland: Springer, 2014, pp. 188–203.

[11] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. ICCV*, Nov. 2011, pp. 263–270.

[12] J. Ahmed, A. Ali, and A. Khan, "Stabilized active camera tracking system," *J. Real-Time Image Process.*, vol. 11, no. 2, pp. 315–334, 2016.

[13] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.

[14] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. ICCV*, Dec. 2015, pp. 4310–4318.

[15] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1090–1097.

[16] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, Nottingham, U.K., Sep. 2014, pp. 1–11.

[17] 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.

[18] H. K. Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4630–4638.

[19] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. ECCV*. Berlin, Germany: Springer, 2012, pp. 702–715.

[20] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. ECCV*. Cham, Switzerland: Springer, 2014, pp. 254–265.

[21] 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. 2017.

[22] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 353–361.

[23] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proc. ICCV*, Dec. 2015, pp. 3101–3109.

[24] J. Yang, Y. Zhu, B. Jiang, L. Gao, L. Xiao, and Z. Zheng, "Aircraft detection in remote sensing images based on a deep residual network and super-vector coding," *Remote Sens. Lett.*, vol. 9, no. 3, pp. 228–236, 2018.

[25] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. ICCV*, Dec. 2015, pp. 3074–3082.

[26] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, no. 2, Jul. 2017, pp. 6931–6939.

[27] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4819–4827.
[28] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3786–3795.
[29] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. ECCV*, 2016, pp. 472–488.
[30] D. Huang, L. Luo, Z. Chen, M. Wen, and C. Zhang, "Applying detection proposals to visual tracking for scale and aspect ratio adaptability," *Int. J. Comput. Vis.*, vol. 122, no. 3, pp. 524–541, 2016.
[31] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4902–4912.
[32] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5388–5396.
[33] A. Ali, A. Jalil, J. Ahmed, M. A. Iftikhar, and M. Hussain, "Correlation, Kalman filter and adaptive fast mean shift based heuristic approach for robust visual tracking," *Signal, Image Video Process.*, vol. 9, no. 7, pp. 1567–1585, 2015.
[34] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. ICCV*, Jul. 2017, pp. 1135–1143.
[35] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4266–4274.
[36] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with Gaussian processes regression," in *Proc. ECCV*. Cham, Switzerland: Springer, 2014, pp. 188–203.
[37] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1401–1409.
[38] A. Lukežič, T. Vojíř, L. Č. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter tracker with channel and spatial reliability," *Int. J. Comput. Vis.*, vol. 126, no. 7, pp. 671–688, 2016.
[39] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 749–758.
[40] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 127–141.

QIANBO LIU received the M.S. degree from the South China University of Technology, China, in 2004, where he is currently pursuing the Ph.D. degree with the School of Mechanical and Automotive Engineering. His research interests include visual tracking, deep learning, and object detection.

GUOQING HU received the M.S. degree from Northwestern Polytechnical University, China, and the Ph.D. degree from Sichuan University, China. He was a Professor at Xiamen University, China. He was an advanced Visiting Scholar with The Chinese University of Hong Kong and The University of Nottingham. He is currently a Professor and a Ph.D. Student Supervisor with the School of Mechanical and Automotive Engineering, South China University of Technology, China. He has completed and participated more than 90 projects, including the National 863 Project, the National Natural Science Foundation Project, the national major projects, international cooperation projects, provincial key projects, and the province fund cooperation projects. He has published 248 papers, 22 patents, and two textbooks. His research interests include amphibious flying machine, intelligent robot, industrial image processing, automation and industrial robot, electromechanical integration, and advanced sensor technology.

MD MOJAHIDUL ISLAM received the B.Sc. and M.Sc. degrees from Islamic University, Bangladesh. He is currently pursuing the Ph.D. degree with the South China University of Technology, Guangzhou, China. From 2010 to 2015, he was an Assistant Professor with the Department of Computer Science and Engineering, Islamic University. His research interests include computer vision, object tracking, pattern recognition, multimedia analysis, and machine learning.