Occlusion-Aware Real-Time Object Tracking

Xingping Dong, Jianbing Shen, *Senior Member, IEEE*, Dajiang Yu, Wenguan Wang, Jianhong Liu, and Hua Huang

***Abstract*—The online learning methods are popular for visual tracking because of their robust performance for most video sequences. However, the drifting problem caused by noisy updates is still a challenge for most highly adaptive online classifiers. In visual tracking, target object appearance variation, such as deformation and long-term occlusion, easily causes noisy updates. To overcome this problem, a new real-time occlusion-aware visual tracking algorithm is introduced. First, we learn a novel two-stage classifier with circulant structure with kernel, named integrated circulant structure kernels (ICSK). The first stage is applied for transition estimation and the second is used for scale estimation. The circulant structure makes our algorithm realize fast learning and detection. Then, the ICSK is used to detect the target without occlusion and build a classifier pool to save these classifiers with noisy updates. When the target is in heavy occlusion or after long-term occlusion, we redetect it using an optimal classifier selected from the classifier-pool according to an entropy minimization criterion. Extensive experimental results on the full benchmark demonstrate our real-time algorithm achieves better performance than state-of-the-art methods.**

***Index Terms*—Circulant structure, classifier-pool, entropy minimization, occlusion, real-time, visual tracking.**

I. Introduction

VISUAL object tracking is an important problem in intelligent surveillance. It has been applied to many applications, especially for human-computer interaction, security and surveillance, and auto-control systems [1]–[7], [9], [11], [15], [16]. It is still a challenging problem caused by several factors, such as illumination variations, occlusion, shape deformation, scale variation, and background clutter. In this paper, we try to solve some of these issues, especially for occlusion.

Existing tracking algorithms can be roughly classified as two categories, namely generative and discriminative methods. The generative methods transform the tracking problem as a nearest-neighbor searching task for the target model [14], [17], [18]. In these methods, the target models are constructed by templates or sparse representation models in subspace. The other discriminative methods aim to learn a binary classifier to discriminate the target appearance from the background [19], [20], [42], [44]. This classifier is trained online using sample patches of the target and the surrounding background. In order to detect the target in subsequent frames, the classifier should be applied exhaustively at many locations. Then the detected new patch can be used to update the classifier model. For those discriminative methods, it is important to select positive samples as well as negative samples, and more samples generally make the classifier more robust. Limited by the number of positive samples, it is natural to select more negative samples from different locations and scales. However, due to the time-sensitive fact of tracking, only a few samples are randomly chosen from each frame for many methods [10], [21]–[24].

Recently, some online learning algorithms with circulant structure [12], [25] are proposed for visual tracking. These algorithms make the classifier models more robust by using thousands of samples for training. These samples are constructed by circulant matrices, which provide a useful connection between the popular learning algorithms and the Fourier domain. It means that the classical Fast Fourier Transformation (FFT) can be applied to solve the learning problems. This transformation will significantly improve the processing speed of the original learning algorithms. In fact, the earliest tracking algorithm with circulant structure with kernel (CSK) [25] can handle hundreds of frames per second (fps) for general tracking tasks. Danelljan *et al.* [12] add the adaptive color name features into CSK tracker, named by CSK-CN, to get better performance with the average speed of more than 100 fps. Although these tracking algorithms have high speed and perform well for many videos, they cannot avoid the drifting problem in the case of noisy updates, which is a general problem for most highly adaptive online classifiers [26]. In visual tracking, these noisy updates may be caused by significant deformation or heavy occlusion. In this paper, we pay more attention to solve this problem. For example, in Fig. 1(b), the CSK-CN losses the target after long-term occlusion, while our tracker successfully re-detects the target.

There exist several algorithms are able to handle partial and heavy occlusion [8], [36], [37], [39], [41], [23], [10], [30]. In [36], a target template is divided into several small cells. They decide whether the cells are occluded using the ratio of its affinities with the neighboring background template and the affinity with target template. Holzer *et al.* [37] use the mean image intensity difference between two consecutive frames and a threshold to detect occlusion. Hare *et al.* [23] propose a structured output support vector machine to estimate the target's location directly, instead of an intermediate classification step. In [10], a tracking-learning-detection (TLD) framework is proposed for long-term tracking. To avoid drifting problem, they develop a P-N learning method to estimate the tracker's errors caused by noisy updates

Manuscript received October 21, 2015; revised February 7, 2016, April 29, 2016, July 18, 2016, and September 27, 2016; accepted November 11, 2016. Date of publication November 22, 2016; date of current version March 15, 2017. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2013CB328805, in part by the National Natural Science Foundation of China under Grant 61272359, and in part by the Fok Ying-Tong Education Foundation for Young Teachers, Specialized Fund for Joint Building Program of Beijing Municipal Education Commission. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhen Wen. *(Corresponding author: Jianbing Shen.)*

The authors are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenjianbing@bit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2016.2631884

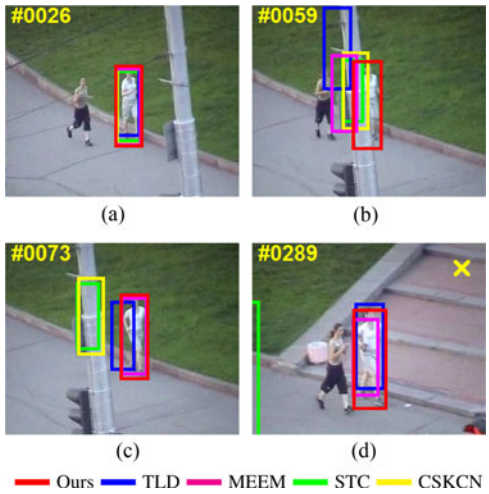

Fig. 1. Comparisons with state-of-the-art trackers on the challenging Jogging sequence [3], including out-of-plane rotation, significant deformation, and long-term occlusion. Our tracker uses large number of samples to train the classifier and performs more robustly to significant deformation than the TLD [10]. For redetecting the target after long-term occlusion, our tracker performs better than the CSKCN [12] and the STC [13] after the 59th frame, since we use the classifier-pool to avoid the drift problem. Our tracker is also faster than the TLD [10] and the MEEM [30] to redetect the target.

and correct them. Zhang *et al.* propose a multi-expert restoration scheme to address the drift problem. They use a minimum entropy criterion to select the best expert to correct undesirable model updates. Overall, these algorithms show effectiveness for occlusion. However, they are less robust for the videos without occlusion compared to the methods using circulant structure, since only a few samples for each frame are considered. For example, the TLD and MEEM methods deviate the position of target in Fig. 1(d).

In order to handle the videos with heavy occlusion and exploit as many samples as possible, we propose a real-time occlusion-aware visual tracking algorithm with kernelized circulant structure. Firstly, we learn a novel classifier with circulant structure, named as Integrated Circulant Structure Kernels (ICSK), which is used to estimate both transition and scale. For each frame, we set an ICSK trained in previous frame as a current classifier to predict the target. The ICSK classifier can provide robust results for no occlusion or partial occlusion and own high handling speed. However, it may be invalid for frames with heavy occlusion. Thus, we build a short-term classifier-pool to store the last K ICSK classifiers without occlusion and corresponding templates (target patches) to handle this situation. If the target is completely occluded, an optimal classifier is selected from the classifier-pool instead of the current classifier according to an evaluation function of classifier, which is based on entropy minimization. This optimal classifier only contains the target information. Therefore, it effectively avoids the drifting problem and can be used to re-detect the target after long-term occlusion. Combined the current ICSK classifier with the classifier-pool, we design a new tracking system to achieve more robust tracking performance. Fig. 2 gives the flowchart of our system and shows the different strategies according to different situations: without occlusion, partial occlusion, and complete occlusion. The main three processing steps for each new coming frame include the pre-update, prediction and update, respectively corresponding to the red, green and blue arrows. In step 1 (pre-update), the templates from classifier-pool are used to discriminate the occlusion by comparing the distances between the previous target patch and them. Only if the previous target is completely occluded, the current ICSK classifier will be updated by the optimal classifier from the classifier-pool. In step 2 (prediction), the features of input frame are extracted to predict the target by using the updated current classifier. According to the location of target, we can get the new training samples for the next frame. In step 3 (update), the current classifier is updated by using the training samples. We use the templates from the classifier-pool to discriminate the occlusion of the predicted target by comparing their distances. If the predicted target is not occluded, the classifier-pool will also be updated. In addition, we propose a new feature by combining the color name feature (CN) with the histogram of orientation gradient (HOG) to further improve the tracking performance. Our source code will be available.[1]

The main contributions are summarized as follows.

1) A novel tracking framework with circulant structure kernels (CSK) is proposed to handle the videos with heavy occlusion, by using varying strategies for each frame with different situations: without occlusion, partial occlusion, and complete occlusion.
2) The proposed algorithm is able to further improve the tracking performance by using combined features CN and HOG and keep real-time processing speed through applying the Fast Fourier Transformation to exploit as many training samples as possible.

II. Related Work

Our algorithm is not only a discriminative tracking method, but also belongs to the tracking-by-detection methods. Next, we will briefly introduce some popular tracking-by-detection methods in recent years. A comprehensive review can be referred to very recent surveys [27]. We further offer detailed discussions about the most related tracking methods with circulant structure, which are substantially the ones with correlation filtering in some special conditions [43]. Therefore, some correlation filtering algorithms are also introduced.

A. Tracking-by-Detection

Many tracking-by-detection algorithms are based on the classifiers, such as Support Vector Machines (SVM) [19], Random Forest classifiers [21] or boosting variants [22], and they are adapted for online learning. Inspired by compressive sensing techniques, Zhang *et al.* [24] propose non-adaptive random projections to extract features from the multi-scale image feature space with data-independent basis. These features are further utilized to train a naive Bayes classifier with online update for real-time tracking. In [28], a robust appearance model containing both holistic templates and local representations is proposed to handle drastic appearance change. This appearance model is used to build a sparsity-based discriminative classifier and

[1][Online]. Available: http://github.com/shenjianbing/occlusiontracking

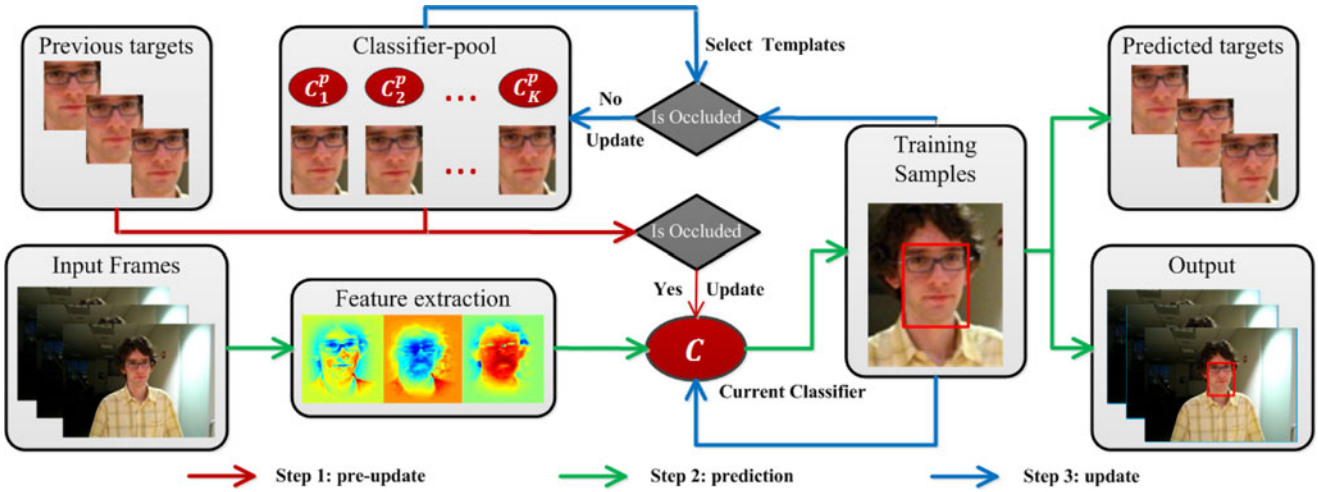

Fig. 2. Framework of our tracking algorithm. The input includes the current frame and a predicted target in previous frame. The previous target is used to discriminate the occlusion. The patch *feature extraction* shows the color maps of a gray feature and two color name features. Red ellipses represent the proposed ICSK classifiers, which contain transition and scale estimations. First, the current classifier will be pre-updated if the target in previous frame is completely occluded. Second, the updated classifier is applied to predict the target in current frame. Third, we update the current classifier by using the training samples. If the predicted target is not occluded, the classifier-pool will be updated.

a sparsity-based generative model. Then the authors combine these two models to deal with appearance change effectively and alleviate the drift problem. Similarly, Chen *et al.* [29] design a learning framework for collaborating the descriptive and discriminative components for tracking. The descriptive component with 1-class SVM forms a robust object model, and the other component based on structured output SVM is used to distinguish the object from the current background. Another SVM-based tracking algorithm is proposed in [30], where an online SVM on a budget algorithm is exploited as a base tracker. The tracker for each frame can be viewed as an expert trained by all previous frames. To address the model drift problem in most online tracking, they usually construct an expert ensemble by using the trackers in the current frame and some nearest previous frames. Then the current tracker is replaced with the best expert selected from the expert ensemble according to a minimum entropy criterion. Our purpose is to design a new classifier-pool visual tracking method to process the occlusion. However, our algorithm is different to the expert ensemble in [30]. In our method, only the reliable classifiers in the no-occlusion frames are used to build the classifier-pool. In [30], the expert ensemble is built by classifiers in previous t frames. If the target is occluded with too long time, all classifiers in the expert ensemble may be invalid in their method. In contrast, our method can skip these invalid classifiers in the occluded frames and keep reliable classifiers.

B. Tracking With Circulant Structure

One important advantage of tracking with circulant structure is that it can exploit large number of samples to train the discriminative classifier for better performance. Henriques *et al.* [25] observe that circulant structure is a good choice to exploit large number of samples. Benefited from the well-established theory of circulant matrices, they propose a link to Fourier analysis and realize extremely fast learning and detection using the Fast Fourier Transform (FFT). Moreover, a kernel classifier is provided to handle more sequences and still keep the speed as fast as linear classifiers. The high speed attracts many researchers to study this CSK algorithm in deep. Instead of the illumination intensity features used in CSK, Danelljan *el al.* [12] use adaptive color name features to improve performance. They propose a robust update strategy to exploit all pervious frames and use an adaptive dimensionality reduction technique for color attributes to preserve useful information and provide a significant speed boost. Zhang *et al.* [13] propose a similar algorithm adopting FFT for fast learning and detection. They exploit the dense spatio-temporal context for visual tracking and further propose an explicit scale adaptation scheme to deal with target scale variations. Recently, in Kernelized Correlation Filter (KCF) [43], Henriques *et al.* improve their CSK tracker by using HOG features and provide a high speed which is only slightly lower than CSK. Moreover, they provide a connection to correlation filters which have been applied for a part of signal processing. Correlation filters have also been widely used in many applications such as object detection and tracking [31], [32]. In [31], a minimum output sum of squared error (MOSSE) filter is proposed for visual tracking on gray-scale images. By using the correlation fitters, the speed of MOSSE tracker is extremely fast even far higher than CSK tracker. The DSSCF tracker [32] learns discriminative correlation filters with a scale pyramid representation to handle the scale change of target objects. They learn two kinds of filters, one for translation estimation and the other for scale estimation. However, these methods do not address the drifting problem regarding online model update. We try to overcome this problem for real-time occlusion-aware visual tracking with kernelized circulant structure.

III. Our Approach

As mentioned before, each frame in the video is classified into one of the three situations: without occlusion, partial

occlusion, and complete occlusion. Then, we separately process these situations to solve the occlusion problem in tracking.

A. Tracking Without Occlusion

Some existing algorithms [12], [25], [32] have good performance and real-time speed on the videos without occlusion. We first employ the Circulant Structure Kernels with Color Names (CSK-CN) [12] and the Discriminative Scale Space Correlation Filter (DSSCF) [32] to construct a new Integrated Circulant Structure Kernels (ICSK) to process the frames without occlusion. The ICSK is a two-stage discriminative classifier including transition estimate and scale estimate, which is similar to the process in [32].

In CSK-CN, a classifier $f(x_{m,n}, w) = \phi(x_{m,n}) \cdot w$ is trained on an image patch x of size $M \times N$ which includes the target, where ϕ is the mapping to the Hilbert space induced by a kernel. All cyclic shifts $x_{m,n}$, $(m,n) \in \{0, \cdots, M-1\} \times \{0, \cdots, N-1\}$ are considered as the training examples. The corresponding label $y(m,n)$ is produced from a Gaussian function y. The classifier is trained by minimizing the following cost function:

$$w = \arg\min_{w} \sum_{m,n} \|\phi(x_{m,n}) \cdot w - y(m,n)\|^2 + \lambda \|w\|^2 \quad (1)$$

where λ is the regularization parameter ($\lambda > 0$). According to the property of circulant matrices [25], the Fast Fourier Transformation (FFT) is applied to minimize the cost function. The solution is defined as $w = \sum_{m,n} a(m,n)\phi(x_{m,n})$, and the coefficient a is formulated as

$$\mathcal{F}(a) = \frac{\mathcal{F}(y)}{\mathcal{F}(\phi(x) \cdot \phi(x)) + \lambda} \quad (2)$$

where $\mathcal{F}$ is the discrete Fourier operator and $y = \{y(m,n) | (m,n) \in \{0, \cdots, M-1\} \times \{0, \cdots, N-1\}\}$. Then the classifier is applied for an image patch z of size $M \times N$ in the new frame to compute the response map as

$$\hat{y} = \mathcal{F}^{-1}(\mathcal{F}(a) \odot \mathcal{F}(\phi(z) \cdot \phi(\hat{x}))) \quad (3)$$

where $\hat{x}$ is the learned target appearance, and $\odot$ is the Hadamard product.

Therefore, the predicted position is the location with maximum score in this response map $\hat{y}$. This position $\mathbf{p}$ is formulated as follows:

$$\mathbf{p} = \mathbf{p}_f - \left(\left\lfloor \frac{M}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor \right) + \arg\max_{(m,n)} \hat{y}(m,n) \quad (4)$$

where $\mathbf{p}_f$ is the target center position at previous frame, and $\lfloor \cdot \rfloor$ means to take the rounded-down integer.

The corresponding maximum response score R is defined as follows:

$$R = \max_{(m,n)} \hat{y}(m,n). \quad (5)$$

A robust update strategy considering all pervious frames is employed to compute the current classifier and appearance in CSK-CN [12]. They update the models with the learning rate γ. For more details, we refer readers to [12].

Given the predicted position for current frame, the DSSCF [32] is used for scale estimation. The target size in the current frame is set as $M_T \times N_T$, then S image patches are extracted around this predicted position. The size of each patch J_n is $\alpha^n M_T \times \alpha^n N_T$, where $n \in \{\lfloor -\frac{S-1}{2} \rfloor \lfloor -\frac{S-3}{2} \rfloor \cdots, \lfloor \frac{S-1}{2} \rfloor\}$, and α denotes the scale factor. All S patches are resized to target size $M_T \times N_T$ for the feature extraction. The final scale estimation result S_r is given as the patch with the highest filtering response. Similar to CSK-CN, the learning rate μ updates the model parameters in an interpolating manner.

B. Tracking With Occlusion

The proposed ICSK may fail to track the target with heavy occlusion. That is because the updating strategy may add some incorrect information into the classifier when the target is occluded. Some incorrect samples (predicted target patches) caused by occlusion are used to update the classifier. With the time of occlusion increasing, the classifier will contain more noisy information and lose discriminability gradually. To solve this problem, we propose a novel updating strategy to track targets with occlusion. The proposed ICSK first tracks the target without occlusion. For each frame t, we get a classifier C_t and a target patch. Since classifiers in the frames far from the current frame are not very efficient to construct the current classifier, we only save the classifiers and target patches in the previous K frames, which are without occlusion and near to the current frame, to construct a classifier-pool. The structure of this pool is a queue, i.e., when a new frame without occlusion is coming in, the frame in the head of queue is left out. For the frames without occlusion, we set the classifier in the end of pool as the current classifier, which is used to predict the position and scale of target in next frame.

When the target is occluded, how to update the current classifier is our main concern. Assume that the occlusion begins at the t-th frame, we stop updating the classifier-pool from the t-th frame until the occlusion is over. There are two situations for occlusion: partial occlusion, and complete occlusion. For partial occlusion, we only use ICSK to update the current classifier since ICSK performs well for most partial occlusion. A target with complete occlusion will experience three stages. In the first stage, the target is approaching the occluded objects, and the current classifier is updated by ICSK since it still belongs to partial occlusion. In the second stage, the target is completely or heavily occluded. We select an optimal classifier from the classifier-pool as the current classifier. The classifier at the end of the first stage has accumulated enough information of the occluded objects, and it will become a better classifier to recognize them. Therefore, if we still use ICSK to update this classifier, the tracking box may stay at the occluded objects. To avoid this situation, we need to replace the current classifier by a more correct classifier selected from the classifier-pool. In the last stage, the target is leaving from the occluded objects. We then use ICSK to update the classifier at the end of the second stage, since the target also belongs to partial occlusion. Next, we will discuss two important questions in details: how to

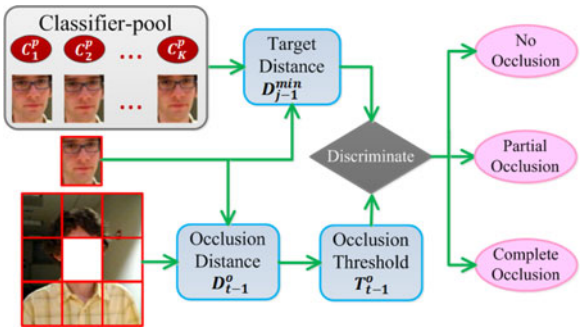

Fig. 3. Discriminating occlusion. The left-middle patch is the target patch in frame t. The left-bottom eight patches are the corresponding surrounding patches, which are used to compute the *occlusion threshold*. The classifier-pool in the left-top is applied to calculate the *target distance*. Then the *target distance* and *occlusion threshold* are used to discriminate the occlusion.

discriminate the occlusion and how to select the optimal classifier from classifier-pool.

1) Discriminating Occlusion: Two distance measures are introduced for identifying occlusion of target patch. As shown in Fig. 3, we define the first distance as *occlusion threshold* by computing the occlusion distance between the target patch and its surrounding patches. This distance is based on the fact that the target object is occluded by the objects surround itself, which is updated frame by frame according to the historical occlusion distances. The second distance is defined as *target distance*, which is the distance between the target patch and patches in the classifier-pool. We only compare these two distances, then we can discriminate whether the target is occluded or not.

Before giving more details, we first determine the distance measure between two patches. A simple idea is to use the squared Euclidean distance between the normalized vectorization of patches. However, as mentioned in [18], this distance metric is not very accurate and robust. It would benefit more for the brighter local parts, i.e., a small variation of distance in brighter local parts would cause a large variation in the whole patch. This situation should be avoided since a brighter local part is not more significant or discriminative than other parts. To assign uniform weights to local parts, we compute a Local HOG Distance (LHD) to measure the center patch and the others. Each patch is first subdivided into n_{local} local patches with size $S_{local} \times S_{local}$. For each local patch, a uniform Histogram of Oriented Gradient (HOG) with 9 directions is used as the feature. In this paper, we set $S_{local} = 8$. Then the feature of a whole patch is defined as a matrix $d_{hog} \times n_{local}$. Let $\mathbf{V}_1$, $\mathbf{V}_2 \in \mathbb{R}^{d_{hog} \times n_{local}}$ as the features of two patches respectively, the Local HOG Distance is defined as follows:

$$D(\mathbf{V}_1, \mathbf{V}_2) = \frac{1}{n_{local}} Tr((\mathbf{V}_1 - \mathbf{V}_2)^T (\mathbf{V}_1 - \mathbf{V}_2)) \quad (6)$$

where $Tr(\cdot)$ represents the trace of a matrix.

Given the distance measure, the proposed two distances (*occlusion threshold* and *target distance*) are defined as follows. Firstly, for each frame t, we compute the LHDs between the center patch and its surrounding patches, and then select the minimal distance D_t^o to compute the *occlusion threshold* T_t^o

$$T_t^o = \begin{cases} (1-\nu)T_{t-1}^o + 0.95\nu D_t^o, & t \text{ without occlusion} \\ T_{t-1}^o, & \text{otherwise} \end{cases} \quad (7)$$

where ν is the learning rate. We empirically use 0.95 ν to make the occlusion detection more robust.

Secondly, we compute the LHDs between the target patch from the previous frame and the target patches from the classifier-pool, and define the minimal distance as *target distance* ($D_{t-1}^{\min}$). Armed with these two distances, we discriminate the target patch with following three situations: complete occlusion, partial occlusion, and no occlusion. If $D_{t-1}^{\min} > T_{t-1}^o$, the target in frame t is completely occluded. If $D_{t-1}^{\min} \leq T_{t-1}^o$, the target is partially occluded or not occluded. Similarly, the minimal distance $D_t^{\min}$ in frame t is computed to discriminate the last situation and the others. If $D_t^{\min} < \eta T_{t-1}^o$, the target is not occluded. If $D_t^{\min} \geq \eta T_{t-1}^o$, the target is partially or completely occluded. η is an adaptive parameter to adjust the threshold of partial occlusion.

The parameter η is set according to different situations. The targets in some videos may have large variation between two adjacent frames, caused by fast motion or deformation. For these videos, the distance $D_t^{\min}$ of the frame without occlusion may be larger. So the parameter η should also be set larger. However, some targets may change a little between two adjacent frames. Correspondingly, the parameter η should be smaller, and η is adjusted by the maximal responding scores of the classifiers in the first K^η frames. Denote R_t as the responding score of frame t. If the current response R_t is large or the distance of adjacent responses $|R_t - R_{t-1}|$ is small, the target will have slight variation in a short period or between two adjacent frames. Then the distance $D_t^{\min}$ will be small, and the parameter η should be set as a small value. For $t = 2, 3, \cdots, K^\eta$, if $|R_t - R_{t-1}| < 0.2$ and $R_t > 0.3$, then we set $\eta = 0.5$; otherwise, we set $\eta = 0.8$. We set K^η as 10. All these values are empirically set according to our experiments.

2) Selecting the Optimal Classifier: An energy function E is proposed to measure the performance of classifiers $\{C_{t,k}^p\}_{k=1}^K$ in the classifier-pool of frame t. This energy function is defined on probability model, and we first give the probability interpretation of a classifier. For simplification, the notation $C_{t,k}^p$ is denoted as C_k^p in this subsection. Each classifier C_k^p can be viewed as a non-parameter distribution and the corresponding response $R_k^p(x_{m,n})$ is viewed as the likelihood of a sample $x_{m,n}$. Given two kinds of labels: l_1 (the target sample) and l_2 (the non-target sample). Then the corresponding normalized likelihood is defined as

$$P_k(l_1|x_{m,n}) = \begin{cases} R_k^p(x_{m,n}), & \text{if } 0 \leq R_k^p(x_{m,n}) \leq 1 \\ 0, & R_k^p(x_{m,n}) < 0 \\ 1, & R_k^p(x_{m,n}) > 1 \end{cases} \quad (8)$$

and $P_k(l_2|x_{m,n}) = 1 - P_k(l_1|x_{m,n})$.

According to this definition, the energy function is formulated as follows:

$$E(C_k^p) = -L(\mathbf{x}; C_k^p) + \lambda_h H(\mathbf{l}|\mathbf{x}; C_k^p) \quad (9)$$

where $\mathbf{x} = \{x_{m,n} | (m,n) \in \{0, \cdots, M-1\} \times \{0, \cdots, N-1\}\}$ is the set of all cyclic shifts in an image patch, $\mathbf{l} = \{l_1, l_2\}$ is the set of labels, the log likelihood is defined as

$$L(\mathbf{x}; C_k^p) = \max_{(m,n)} \log P_k(l_1 | x_{m,n}) \tag{10}$$

and the entropy term is computed by

$$H(\mathbf{l}|\mathbf{x}; C_k^p) = -\frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{i=1}^{2} \times P_k(l_i | x_{m,n}) \log P_k(l_i | x_{m,n}). \tag{11}$$

The first term in (9) is used to keep the consistence of the energy function and the responding score of the classifier. The larger the maximal response is, the better the classifier is, and the corresponding energy should be smaller. The second term in (9) is an entropy regularization term which has been applied for some learning [33] and tracking tasks [30]. This term favors the distribution that the classifier with low ambiguity. For example, a sample is assigned to label l_1 or label l_2. For this sample, a classifier having a high confidence score to one label and a low confidence score to the other is more favored than the one having similar confidences of both labels.

Finally, the optimal classifier $C_{k^*}^p$ is selected from the classifier-pool by the following function:

$$k^* = \arg \min_{k=1}^{K} E(C_k^p) \tag{12}$$

where the energy $E(C_k^p)$ of each classifier C_k^p is computed as mentioned before. Via this function, the one minimal enenrgy is selected as the optimal classifier $C_{k^*}^p$.

The tracking algorithm for each frame is summarized with pseudo-code in Algorithm 1.

Algorithm 1: Occlusion-aware tracking at time step t

Input: Frame t. Previous target position $\mathbf{p}_{t-1}$, scale s_{t-1}, maximal response R_{t-1}, and target patch feature $\mathbf{V}_{t-1}$. ICSK model C_{t-1}, occlusion threshold T_{t-1}^o, classifier-pool $\{C_{t-1,k}^p\}_{k=1}^K$, and patch-pool $\{\mathbf{V}_{t-1,k}^p\}_{k=1}^K$;

Output: Estimated target position $\mathbf{p}_t$, scale s_t, maximal response R_t, and target patch feature $\mathbf{V}_t$. Updated: ICSK model C_t, occlusion threshold T_t^o, classifier-pool $\{C_{t,k}^p\}_{k=1}^K$, and patch-pool $\{\mathbf{V}_{t,k}^p\}_{k=1}^K$;

Compute distance $D_{t-1}^{\min} = \min_{k=1}^{K} D(\mathbf{V}_{t-1}, \mathbf{V}_{t-1,k}^p)$.
if $D_{t-1}^{\min} > T_{t-1}^o$ (complete occlusion) **then**
  Select the optimal classifier C_{t-1,k^*}^p via eq. 12;
  Update: $C_{t-1} \leftarrow C_{t-1,k^*}^p$, $\mathbf{V}_{t-1} \leftarrow \mathbf{V}_{t-1,k^*}^p$;
  Compute $\mathbf{p}_t$ and R_t using C_{t-1}, and $s_t \leftarrow s_{t-1}$;
else
  Compute $\mathbf{p}_t$, s_t and R_t using C_{t-1};
end if
Extract the appearance model $\hat{x}$ and $\mathbf{V}_t$ using $\mathbf{p}_t$, s_t;
Train the ICSK C_t with $\hat{x}$;
if $t < K^\eta$ **then**
  if $|R_t - R_{t-1}| < 0.2$ and $R_t > 0.3$ **then** $\eta = 0.5$;
  else $\eta = 0.8$;
  end if
end if
Compute distance $D_t^{\min} = \min_{k=1}^{K} D(\mathbf{V}_t, \mathbf{V}_{t-1,k}^p)$;
if $D_t^{\min} < \eta T_{t-1}^o$ (without occlusion) **then**
  Update $\{C_{t,k}^p\}_{k=1}^K$, $\{\mathbf{V}_{t,k}^p\}_{k=1}^K$ by adding C_t, $\mathbf{V}_t$;
  Extract patches $\{\mathbf{V}_k^{'}\}_{k=1}^8$ surrounding position $\mathbf{p}_t$;
  Compute distance $D_t^o = \min_{k=1}^{8} D(\mathbf{V}_t, \mathbf{V}_k^{'})$;
  Update threshold $T_t^o = (1-\nu) T_{t-1}^o + 0.95 \nu D_t^o$;
else
  $T_t^o = T_{t-1}^o$;
end if

IV. Experimental Results

In this section, we evaluate the proposed method by comparing with other state-of-the-art visual tracking algorithms. These algorithms include both tracking methods with correlation filter and other popular ones. The correlation-filter-based tracking algorithms include CSK [25], CSK-CN [12], STC [13], DSSCF [32], and KCF [43]. The other state-of-the-art algorithms are CT [24], DFT [34], L1APG [35], SCM [28], LOT [17], TLD [10], PCOM [38], Struct [23], [40], and MEEM [30]. To demonstrate the effectiveness of our tracking algorithm, we evaluate these algorithms on 56 challenging video sequences. The 50 sequences of them are taken from a recent online object tracking benchmark [3]. The residual 6 sequences are collected by Danelljan *et al.* [12] from other datasets. These video sequences span a wide degree of difficulty, such as motion blur, illumination changes, scale variation, heavy occlusions, in-plane and out-of-plane rotations, out of view, background clutter and low resolution.

We combine adaptive color names (CN) and HOG as the features (CN-HOG) for all images. First, we train a classifier for one feature on a frame, respectively. Second, we calculate the responses of these two classifiers on the next frame. Third, the weighted average of these responses is computed as the final response, which predicts the location of target. The weights of CN and HOG features are defined as $\beta \in [0,1]$ and $1-\beta \in [0,1]$, respectively. We set the CN weight as $\beta = 0.9$ through experiments. The other main parameters include three aspects, parameters in ICSK-CN, parameters in DSSCF, and parameters for handling occlusion. In CSK-CN, the main parameters include the regularization parameter $\lambda = 0.01$, the learning rate $\gamma = 0.03$, and the Gaussian kernel with bandwidth $\sigma_g = 0.25$. In DSSCF, the main parameters are set as $S = 29$, $\alpha = 1.015$, and $\mu = 0.025$. The residual important parameter of padding size is set to 1.5. For handling occlusion, the parameters are set as follows. The learning rate of occlusion threshold is set as $\nu = 0.015$, the size of classifier-pool is set as $K = 5$, and the controlling parameter in entropy minimization is set as $\lambda_h = 10$. To accelerate the processing speed, we scale the targets in some sequences to be smaller. If the size of target is large than 4000

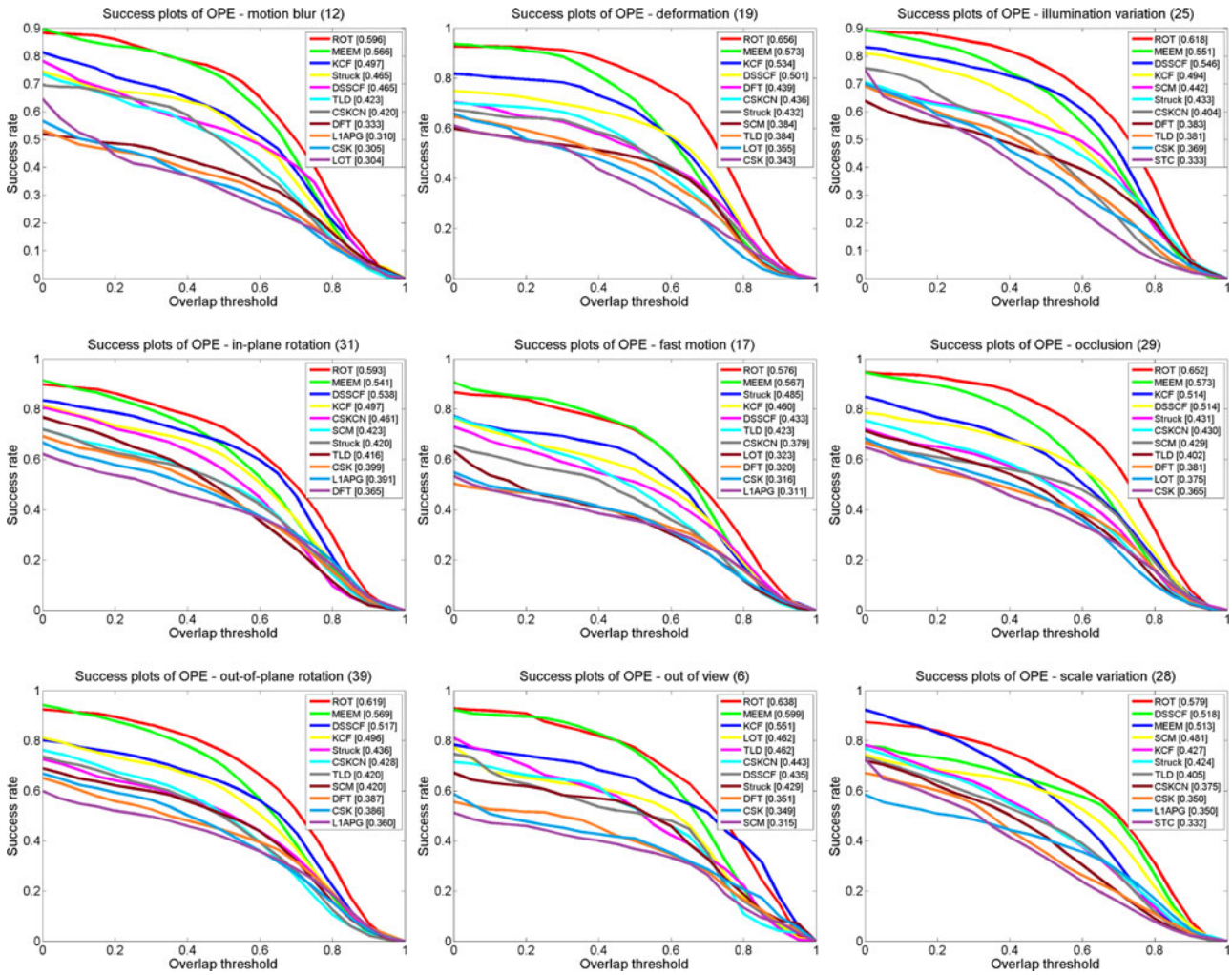

Fig. 4. Overlap success plots over nine tracking challenges of fast motion, background clutter, motion blur, deformation, illumination variation, occlusion, out-of-plane rotation, out-of-view, and scale variation. The legend contains the AUC score for each tracker.

($M_T N_T > 4000$), we then scale this size to 4000. The implementations of CSK [25], CSKCN [12], STC [13], PCOM [38], DSSCF [32], KCF [43], MEEM [30], and Struck [40] are provided by the authors with suggested parameters. And the other implementation algorithms are directly taken from the benchmark [3].

To measure the quantitative performance, we use the following measures from OTB [3], the center location error (CLE), distance precision (DP), overlap success (OS) rate, and speed in terms of frames per second (fps). Here, we just give short introduction for these evaluation metric, and more details are referred to [3]. The center location error is defined as the average Euclidean distance between the manually labeled ground truths and the center locations of the tracked targets. Given a threshold distance, the distance precision is defined as the percentage of frames whose CLE is less than this threshold. The overlap success rate is based on the bounding box overlap. Similar with DP, the overlap success rate also needs a specific threshold and it is represented as the ratio of successful frames whose overlap is larger than the given threshold.

Overall comparison: We now evaluate our method and other well-known approaches on all 56 challenging sequences. We report the distance precision, the overlap success, the center location error and the running time. The results for nine main challenging attributes are reported in Fig. 4. The average computing results for each method are shown in Table I. The first and second best results are shown by bold and underline in each metric, respectively. Our ROT approach significantly improves the baseline CSKCN tracker with a relative reduction in the average CLE from 58.8 to 19.4. Moreover, compared with the baseline method, our ROT tracker improves the average DP from 66.1% to 78.9%. And the average OS is also improved from 55.3% to 72.1%. Among all the trackers, our ROT method achieves the best results with CLE of 19.4 pixels, and it still achieves real-time performance with 29 fps. In our method, the main time cost is spent for scale estimation.

Robustness evaluation: It may be not enough for tracker evaluation to only use one DP or OS at a specific threshold. We further use precision plot and success plot to report the average precision and success rate, respectively. A conventional way to evaluate trackers is one-pass evaluation (OPE), via running these trackers throughout a test sequence with initialization from the ground truth position in the first frame. However some trackers may be sensitive to the initialization. Therefore, to measure

TABLE I
AVERAGE RESULTS ON 56 CHALLENGING SEQUENCES. THE RESULTS ARE REPORTED IN AVERAGE CENTER LOCATION ERROR (CLE), DISTANCE PRECISION (DP), OVERLAP SUCCESS (OS) RATE, AND SPEED (FPS). THE FIRST AND SECOND BEST RESULTS ARE SHOWN BY BOLD AND UNDERLINE

	CT	TLD	Struck	DFT	LOT	L1APG	SCM	CSK	CSKCN	STC	PCOM	DSSCF	KCF	MEEM	ROT-SF	ROT (ours)
CLE	87.1	52.0	46.5	72.2	61.5	76.2	57.8	83.7	58.8	81.1	86.8	43.9	36.0	<u>19.6</u>	34.7	**19.4**
DP	0.355	0.588	0.688	0.481	0.496	0.497	0.605	0.555	0.661	0.525	0.457	0.715	0.724	<u>0.840</u>	0.760	**0.879**
OS	0.239	0.507	0.588	0.433	0.398	0.445	0.558	0.467	0.553	0.349	0.397	0.660	0.617	<u>0.723</u>	0.684	**0.791**
FPS	47.8	29.0	12.8	11.4	0.6	1.1	0.2	<u>325.8</u>	180.2	**367.5**	7.3	32.6	252.8	15.8	35.0	29.0

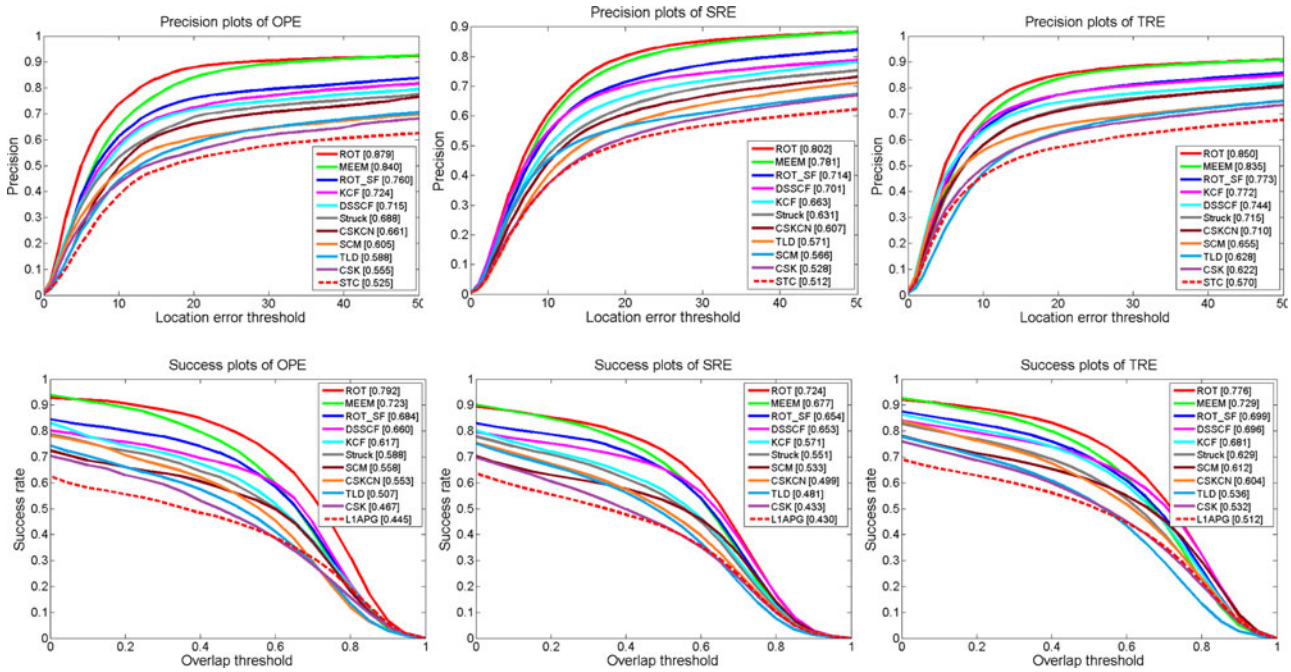

Fig. 5. Plots of OPE, SRE, and TRE on all 56 sequences. The performance score for each tracker is shown in the legend.

the performance with different initialization, we use temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE). TRE starts at different frames and SRE uses different bounding boxes in first frame for initialization. The setup of these two evaluation measures is similar to that in [3]. For precision plots, we rank the trackers by using the results at error threshold of 20, while for success plots we use the area under curve (AUC) of each success plot for ranking. The results in Fig. 5 show that our algorithm achieves better performance than other trackers in terms of OPE, SRE and TRE. Note that the baseline method CSK-CN achieves higher TRE performance compared with OPE, since the long-term occlusion will be reduced or avoided in shorter video sequences and the drifting problem will be less noticeable. Although the TRE does not fully reflect the merits of our approach in handling the drifting problem, the proposed algorithm still outperforms against state-of-the-art tracking methods.

V. CONCLUSION

We have proposed an effective occlusion-aware algorithm for real-time visual tracking with 29 fps. Our method learns discriminative classifiers for estimating the translation and scale variations of target objects efficiently. The translation is estimated by modeling a classifier with kernelized circulant structure. And the scale is estimated by constructing a correlation filter with multi-scale samples to achieve robust tracking results. We further develop a classifier-pool to re-detect targets with an entropy minimization criterion in case of tracking failure. Extensive experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods in terms of accuracy and robustness.

REFERENCES

[1] D.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "Visual tracking using pertinent patch selection and masking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 3486–3493.
[2] J. Shen, X. Yang, Y. Jia, and X. Li, "Intrinsic images using optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 3481–3487.
[3] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 2411–2418.
[4] L. Xie *et al.*, "Tracking large-scale video remix in real-world events," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1244–1254, Oct. 2013.
[5] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.

[6] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 3395–3402.
[7] W. Wang, J. Shen, X. Li, and F. Porikli, "Robust video object co-segmentation," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3137–3148, Oct. 2015.
[8] B. Ma *et al.*, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1818–1828, Oct. 2015.
[9] J. Shen, Y. Du, and X. Li, "Interactive segmentation using constrained Laplacian optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1088–1100, Jul. 2014.
[10] 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.
[11] J. Shen, D. Wang, and X. Li, "Depth-aware image seam carving," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1453–1461, Oct. 2013.
[12] 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 Recog.*, Jun. 2014, pp. 1090–1097.
[13] 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.*, 2014, pp. 127–141.
[14] S. Kwak, W. Nam, B. Han, and J. H. Han, "Learning occlusion with likelihoods for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1551–1558.
[15] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015.
[16] X. Dong, J. Shen, and L. Shao, "Sub-Markov random walk for image segmentation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 516–527, Feb. 2016.
[17] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 1940–1947.
[18] D. Wang, H. Lu, Z. Xiao, and M.-H. Yang, "Inverse sparse tracker with a locally weighted distance metric," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2646–2657, Sep. 2015.
[19] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.
[20] Z. Zuo, G. Wang, B. Shuai, L. Zhao, and Q. Yang, "Exemplar based deep discriminative and shareable feature learning for scene image classification," *Pattern Recog.*, vol. 48, no. 10, pp. 3004–3015, 2015.
[21] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Proc. IEEE ICCV Workshops*, Sep.-Oct. 2009, pp. 1393–1400.
[22] 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.
[23] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 263–270.
[24] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 864–877.
[25] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.
[26] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.
[27] A. W. Smeulders *et al.*, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.
[28] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 1838–1845.
[29] D. Chen, Z. Yuan, G. Hua, Y. Wu, and N. Zheng, "Description-discrimination collaborative tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 345–360.
[30] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.
[31] D. S. Bolme, J. R. Beveridge, B. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 2544–2550.
[32] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.
[33] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 529–536.
[34] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 1910–1917.
[35] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 1830–1837.
[36] D. Chen, Z. Yuan, Y. Wu, G. Zhang, and N. Zheng, "Constructing adaptive complex cells for robust visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1113–1120.
[37] S. Holzer, S. Ilic, and N. Navab, "Multilayer adaptive linear predictors for real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 105–117, Jan. 2013.
[38] D. Wang and H. Lu, "Visual tracking via probability continuous outlier model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 3478–3485.
[39] B. Ma, H. Hu, J. Shen, Y. Zhang, and F. Porikli, "Linearization to nonlinear learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4400–4407.
[40] S. Hare *et al.*, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 1, 2016.
[41] B. Ma, L. Huang, and J. L. Shao, "Discriminative visual tracking using tensor pooling," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2411–2422, Nov. 2016.
[42] B. Ma, H. Hu, J. Shen, Y. Liu, and L. Shao, "Generalized pooling for robust object tracking," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4199–4208, Sep. 2016.
[43] 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.
[44] B. Ma *et al.*, "Visual tracking under motion blur," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5867–5876, Dec. 2016.

Xingping Dong, photograph and biography not available at the time of publication.

Jianbing Shen (M'11–SM'12), photograph and biography not available at the time of publication.

Dajiang Yu, photograph and biography not available at the time of publication.

Wenguan Wang, photograph and biography not available at the time of publication.

Jianhong Liu, photograph and biography not available at the time of publication.

Hua Huang, photograph and biography not available at the time of publication.