Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 238971, 13 pages
http://dx.doi.org/10.1155/2015/238971

Research Article

Robust Visual Correlation Tracking

Lei Zhang,[1,2] Yanjie Wang,[1] Honghai Sun,[1] Zhijun Yao,[1] and Shuwen He[3]

[1]*Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China*
[2]*University of Chinese Academy of Sciences, Beijing 100049, China*
[3]*North Automatic Control Technology Research Institute, Taiyuan 030006, China*

Correspondence should be addressed to Lei Zhang; zhanglei8080@126.com

Received 3 June 2015; Accepted 2 September 2015

Academic Editor: Matteo Gaeta

Recent years have seen greater interests in the tracking-by-detection methods in the visual object tracking, because of their excellent tracking performance. But most existing methods fix the scale which makes the trackers unreliable to handle large scale variations in complex scenes. In this paper, we decompose the tracking into target translation and scale prediction. We adopt a scale estimation approach based on the tracking-by-detection framework, develop a new model update scheme, and present a robust correlation tracking algorithm with discriminative correlation filters. The approach works by learning the translation and scale correlation filters. We obtain the target translation and scale by finding the maximum output response of the learned correlation filters and then online update the target models. Extensive experiments results on 12 challenging benchmark sequences show that the proposed tracking approach reduces the average center location error (CLE) by 6.8 pixels, significantly improves the performance by 17.5% in the average success rate (SR) and by 5.4% in the average distance precision (DP) compared to the second best one of the other five excellent existing tracking algorithms, and is robust to appearance variations introduced by scale variations, pose variations, illumination changes, partial occlusion, fast motion, rotation, and background clutter.

1. Introduction

Visual tracking, as a fundamental step to explore videos, is important in many computer vision based applications, such as face recognition, human behavior analysis, robotics, intelligent surveillance, intelligent transportation systems, and human-computer interaction. The objective of visual tracking is to estimate the locations of a target in a video sequence [1–3]. During the tracking process, the state of the target is estimated over time by associating its representation in the current frame with those in previous frames. Though the research on visual tracking algorithms has lasted for decades, visual tracking is still a problem because of the factors such as pose variation, illumination changes, partial occlusion, fast motion, scale variation, background clutter, and so on.

In general, current tracking algorithms can be classified as either generative or discriminative approaches. Generative approaches [4–7] focus on learning an appearance model and formulate the tracking problem as finding the target observation most similar to the learned appearance or with minimal reconstruction error. The models are based on templates or subspace models. However, these generative models do not take the background information into consideration, therefore throwing away some very useful information that can help to discriminate object from background. Different from generative trackers, discriminative methods [8–13] address the tracking problem as a classification problem which differentiates the tracked targets from the backgrounds. They employ both the target and the background information. For example, Avidan [14] proposes a strong classifier based on a set of weak classifiers to do ensemble tracking. Kalal et al. [15] propose a P-N learning algorithm to learn tracking classifiers from positive and negative samples. These methods are also termed as tracking-by-detection [16–18], in which a binary classifier separates the target from background in the continuous frames. In recent years, tracking-by-detection methods have shown to provide excellent tracking performance.

Most current tracking algorithms are only confined to finding out the target location. This implies poor tracking performance in sequences with great scale changes. Several

methods [19–21] that use Scale Invariant Feature Transform (SIFT) features can adapt to object scale variations at low frame-rates, but they are not able to be used in the real-time applications. Tu et al. [19] propose a vehicle tracking approach combining blob based tracking and SIFT features based tracking, which is robust to the size of vehicle. Jiang et al. [20] present a novel algorithm for object tracking based on particle filter and SIFT. Wei et al. [21] propose a SIFT based mean shift algorithm, which can be used for continuous vehicle tracking in complex situations. In this paper, we present an adaptive scale tracking approach using discriminative correlation filters, which can estimate the target scale accurately. One main contribution of this work is to decompose the tracking task into translation and scale estimation. The target translation and scale estimation both work by making use of the kernelized correlation filters. In addition, we adopt a new update scheme online based on the MOSSE [22] tracker, which takes all the previous frames into consideration when computing the current models. Experimental results on challenging video sequences demonstrate the superior performance of our proposed method in robustness and stability against state-of-the-art methods.

The rest of this paper is organized as follows. A brief summary of the most related work is first given in Section 2. The tracking algorithm with kernelized correlation filters is introduced in Section 3. Section 4 describes our proposed approach. Following this, the experimental results are presented with comparisons to state-of-the-art methods on challenging sequences in Section 5. Finally, we conclude this paper in Section 6.

2. Related Work

Visual object tracking has been studied extensively with lots of applications. In this section, we introduce the approaches closely related to our work.

Correlation filters have been used in many applications such as object detection and recognition [23]. Since the operator is readily transferred into the Fourier domain as element-wise multiplication, correlation filters have attracted considerable attention recently to visual tracking due to its computational efficiency. In recent years, the researchers start to bring the correlation filters into the tracking-by-detection methods and gain a great development. Bolme et al. [22] propose to learn a minimum output sum of squared error (MOSSE) filter for visual tracking on gray-scale images, where the learned filter encodes target appearance with update on every frame. Henriques et al. [24] propose a circulant structure of tracking-by-detection with kernels (CSK) method, which uses correlation filters in a kernel space. They propose the first kernelized correlation filter, but the CSK method only builds on single channel features. Generalizations of linear correlation filters to multiple channels have also been proposed [25–27], which allow them to use more modern features such as histogram of oriented gradients (HOG). Henriques et al. [28] propose a kernelized correlation filter (KCF) tracking algorithm, which is further improved by using HOG features. Danelljan et al. [1] propose an adaptive color attributes tracking method, which exploits the color attributes of a target and learns an adaptive correlation filter by mapping multichannel features into a Gaussian kernel space. However, the above methods do not consider the target scale prediction. Recently, Wu et al. [29] perform a comprehensive evaluation of online tracking algorithms. In the evaluation, the CSK tracker is shown to provide competitive performance with the highest speed among ten top trackers. Due to its excellent performance, we base our approach on the CSK tracker.

3. Kernelized Correlation Filters Based Tracking

For the correlation filter based trackers, correlation can be computed in the Fourier domain through Fast Fourier Transform (FFT); and the correlation response can be transformed back into the spatial domain with the inverse FFT. The CSK tracking method explores a dense sampling strategy while showing the process of taking subwindows in a frame induces circulant structure. The CSK tracker learns a regularized least squares' (RLS) classifier of the target appearance from a single image patch, gets the kernelized correlation filter with using the circulant matrices and kernel trick, and localizes the target in a new frame by finding the maximum response of the correlation filter. In this section, we briefly describe the CSK tracker.

3.1. Circulant Matrices. Assume $C(\mathbf{a})$ is an $n \times n$ circulant matrix; then it can be obtained from a $1 \times n$ vector $\mathbf{a}$:

$$C(\mathbf{a}) = \begin{bmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix}. \tag{1}$$

The first column is the transposition of the vector $\mathbf{a}$, the second column is the transposition of the vector that is $\mathbf{a}$ cyclic shifted one element to the right, and so on. For an $n \times 1$ vector $\mathbf{u}$, the product of $C(\mathbf{a})$ and $\mathbf{u}$ represents the convolution of vectors $\mathbf{a}$ and $\mathbf{u}$ [30]; it can be expressed in the Fourier domain as follows:

$$C(\mathbf{a})\,\mathbf{u} = \mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{a})\,\mathcal{F}(\mathbf{u})\right), \tag{2}$$

where $\mathcal{F}^{-1}$ and $\mathcal{F}$ denote the inverse Fourier transform and Fourier transform, respectively.

3.2. The Regularized Least Squares Classification. It has been shown that, in many practical problems, the RLS classifier offers equivalent classification performance compared to the support vector machine (SVM), and the former is implemented easily [31]. The approach uses a single gray-scale image patch $\mathbf{x}$ which is centred around the target to train the classifier. The classifier $f(\mathbf{x})$ has the form $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, and

it is trained by minimizing the cost function (see (3)) over samples $\mathbf{x}_i$:

$$\arg\min_{w} \sum_{i} \left| f\left(\mathbf{x}_i\right) - y_i \right|^2 + \lambda \left\| \mathbf{w} \right\|^2, \tag{3}$$

where y_i is the desired output for $\mathbf{x}_i$ and λ is a regularization parameter.

Mapping the inputs $\mathbf{x}$ to the feature space $\varphi(\mathbf{x})$ with the kernel trick, the kernel is $\kappa(\mathbf{x}, \mathbf{x}') = \varphi^T(\mathbf{x})\varphi(\mathbf{x}')$. Then we can express the solution $\mathbf{w}$ as a linear combination of the inputs [32]:

$$\mathbf{w} = \sum_{i} d_i \varphi\left(\mathbf{x}_i\right), \tag{4}$$

where d_i is the coefficient.

Then the RLS with kernels has the simple closed form solution [31] as follows:

$$\mathbf{d} = \left(\mathbf{K} + \lambda \mathbf{I}\right)^{-1} \mathbf{y}, \tag{5}$$

where $\mathbf{K}$ is the kernel matrix with elements $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{I}$ is the unit matrix, $\mathbf{y}$ is the desired output with elements y_i, and $\mathbf{d}$ is the transformed classifier coefficient vector with elements d_i.

3.3. Fast Target Location Estimation. It has been proved that the kernel matrix $\mathbf{K}$ is circulant if κ is unitarily invariant [24]. We can get (6) from (5) according to the property of circulant matrices:

$$\mathbf{D} = \mathscr{F}\left(\mathbf{d}\right) = \frac{\mathscr{F}\left(\mathbf{y}\right)}{\mathscr{F}\left(\overline{\mathbf{h}}\right) + \lambda}, \tag{6}$$

where $\mathbf{K} = C(\overline{\mathbf{h}})$ and $\overline{\mathbf{h}}$ is the vector with elements $\overline{h}_i = \kappa(\mathbf{x}, \mathbf{x}_i)$.

We complete the target location detection using the interesting image patch $\mathbf{z}$ in a new frame. Then the response of the RLS classifier is $\widehat{\mathbf{y}} = \mathbf{w}^T\mathbf{z} = \sum_i d_i \kappa(\mathbf{x}, \mathbf{z}_i)$, and it can be computed in the Fourier domain as follows:

$$\widehat{\mathbf{y}} = \mathscr{F}^{-1}\left(\mathbf{D}\mathbf{H}\right), \tag{7}$$

where $\mathbf{H} = \mathscr{F}(\mathbf{h})$, where $\mathbf{h}$ is the vector with elements $h_i = \kappa(\widehat{\mathbf{x}}, \mathbf{z}_i)$, where $\widehat{\mathbf{x}}$ represents the target model learned from the previous frame and $\mathbf{z}_i$ is the sample of the image patch $\mathbf{z}$.

The position of the target in a new frame is obtained by finding the position that makes the $\widehat{\mathbf{y}}$ maximum, which means finding the position that maximizes the response of the filter $\mathbf{h}$, and $\mathbf{D}$ and $\mathbf{x}$ are updated as follows:

$$\widehat{\mathbf{D}}^t = (1 - \alpha)\, \widehat{\mathbf{D}}^{t-1} + \alpha \mathbf{D}^t, \tag{8}$$

$$\widehat{\mathbf{x}}^t = (1 - \alpha)\, \widehat{\mathbf{x}}^{t-1} + \alpha \mathbf{x}^t, \tag{9}$$

where α is the learning rate and $\widehat{\mathbf{D}}^t$ and $\widehat{\mathbf{D}}^{t-1}$ denote the updated coefficients at frame t and frame $t-1$, respectively. $\widehat{\mathbf{x}}^t$ and $\widehat{\mathbf{x}}^{t-1}$ denote the updated target model at frame t and frame $t-1$, respectively. $\mathbf{D}^t$ and $\mathbf{x}^t$ denote the coefficients and target model computed from frame t, respectively. For more details, we refer to [24].

4. The Proposed Visual Tracking Algorithm

In this section, we present the adaptive scale tracking method based on the kernelized correlation filters in detail. Recently, Danell et al. [8] propose a scale estimation method based on the MOSSE filter. Inspired from it, we propose a robust correlation tracking approach based on the CSK tracker. Since the scale changes very little between two frames in visual tracking, we can detect the target position using the position kernelized correlation filter firstly and then estimate the target scale using the scale kernelized correlation filter that is learned by using the samples collected from the detected target. In the following subsections, we will introduce a new online update scheme and a scale prediction strategy.

4.1. Online Update Scheme. Since the appearance of the target often changes significantly during the visual tracking, it is necessary to update the target model to adapt to these changes. In the CSK tracker, the model consists of the transformed classifier coefficients and the learned target model. But they are computed only considering the current appearance. This limits the performance because not all the previous frames are considered to compute the current model. However, the MOSSE tracker [22] employs a robust update scheme by considering all previous frames when computing the current model and performs well. Here we adopt the same idea to update the models in our approach. Then we take all the extracted appearances $\{\mathbf{x}^j : j = 1, \ldots, t\}$ of the target from the first frame till the current frame t into consideration in our update scheme. Therefore, the cost function in (3) can be modified as

$$\arg\min_{w} \sum_{j=1}^{t} \left(\sum_{i} \left| f\left(\mathbf{x}_i^j\right) - y_i^j \right|^2 + \lambda \left\| \mathbf{w}^j \right\|^2 \right). \tag{10}$$

Then the coefficients $\mathbf{D}^t$ for the frame t can be computed as follows:

$$\mathbf{D}^t = \frac{\sum_{j=1}^{t} \mathbf{Y}^j \overline{\mathbf{H}}^j}{\sum_{j=1}^{t} \overline{\mathbf{H}}^j \left(\overline{\mathbf{H}}^j + \lambda\right)}, \tag{11}$$

where $\mathbf{Y}^j = \mathscr{F}(\mathbf{y}^j)$ and $\overline{\mathbf{H}}^j = \mathscr{F}(\overline{\mathbf{h}}^j)$, where $\overline{\mathbf{h}}^j$ is the vector with elements $\overline{h}_i^j = \kappa(\mathbf{x}^j, \mathbf{x}_i^j)$.

Then (7) is expressed as

$$\widehat{\mathbf{y}} = \mathscr{F}^{-1}\left(\mathbf{D}^t\mathbf{H}\right). \tag{12}$$

The target appearance $\widehat{\mathbf{x}}^t$ is updated using (9). Here we update the numerator $\mathbf{D}^t_{\text{Num}}$ and the denominator $\mathbf{D}^t_{\text{Den}}$ of $\mathbf{D}^t$ in (11) separately as

$$\mathbf{D}^t_{\text{Num}} = (1 - \alpha)\, \mathbf{D}^{t-1}_{\text{Num}} + \alpha \mathbf{Y}^t \overline{\mathbf{H}}^t, \tag{13}$$

$$\mathbf{D}^t_{\text{Den}} = (1 - \alpha)\, \mathbf{D}^{t-1}_{\text{Den}} + \alpha \overline{\mathbf{H}}^t \left(\overline{\mathbf{H}}^t + \lambda\right). \tag{14}$$

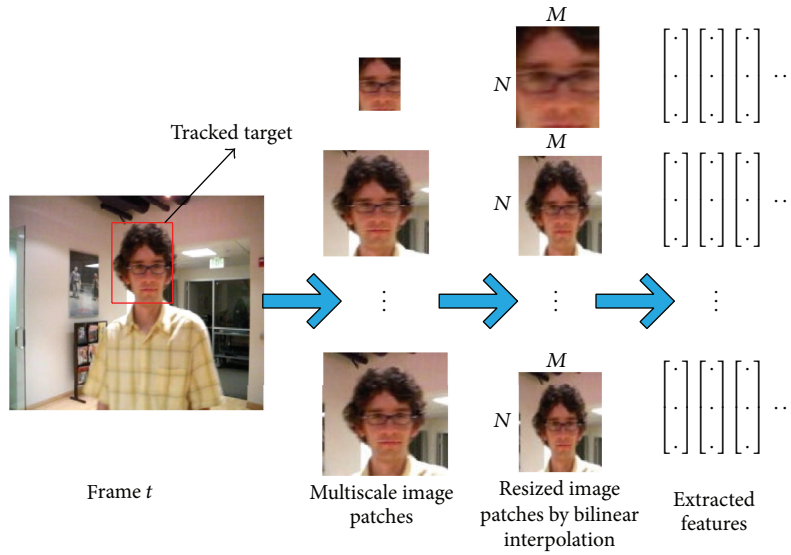


Figure 1: The process of extracting features. We get the multiscale image patches around the tracked target at frame t, then resize the patches to the initial target size $M \times N$ by bilinear interpolation, and extract HOG features from these resized patches.

4.2. The Target Scale Prediction Strategy. To predict the target scale variation, we learn another kernelized correlation filter and train another classifier on multiscale image patches around the most reliable tracked targets. During the tracking, we construct a target pyramid around the tracked target to estimate the target scale. We resize the patches by using the bilinear interpolation to the size of the initial target before extracting features. The training samples for learning the filter are computed by extracting HOG features using the resized patches which are centred around the tracked target. Then the extracted features are multiplied by a Hamming window to reduce the frequency effect of image boundary when using the FFT, as described in [22]. Assume the initial target size in the current frame is $M \times N$ and the size of the scale filter is $S \times 1$; then we extract the sample $\mathbf{x}_{s_i}$ from the image patches of size $m \times n$ which are centred around the target, where $m = r^a M$, $n = r^a N$, $a \in \{\lfloor -((S-1)/2) \rfloor, \ldots, \lfloor (S-1)/2 \rfloor\}$, and r is the scale factor. The process of extracting features is shown in Figure 1. We compute the coefficients $\mathbf{D}_s^t$ by (15) and the response $\hat{\mathbf{y}}_s$ for a new frame by (16), update $\mathbf{D}_s^t$ using (13) and (14), and update the scale model $\hat{\mathbf{x}}_s$ using (9). The target scale in a new frame is obtained by finding the scale that makes $\hat{\mathbf{y}}_s$ maximum:

$$\mathbf{D}_s^t = \frac{\sum_{j=1}^{t} \mathbf{Y}_s^j \overline{\mathbf{H}}_s^j}{\sum_{j=1}^{t} \overline{\mathbf{H}}_s^j \left(\overline{\mathbf{H}}_s^j + \lambda \right)}, \tag{15}$$

where $\overline{\mathbf{H}}_s^j = \mathscr{F}(\overline{\mathbf{h}}_s^j)$, with $\overline{\mathbf{h}}_s^j$ being the vector with elements $\overline{h}_{s_i}^j = \kappa(\mathbf{x}_s^j, \mathbf{x}_{s_i}^j)$, where $\mathbf{x}_s^j$ is the learned scale model from frame $j-1$, and $\mathbf{Y}_s^j = \mathscr{F}(\mathbf{y}_s^j)$, where $\mathbf{y}_s^j$ is the desired output for $\mathbf{x}_s^j$ at frame j:

$$\hat{\mathbf{y}}_s = \mathscr{F}^{-1} \left(\mathbf{D}_s^t \mathbf{H}_s \right), \tag{16}$$

where $\mathbf{H}_s = \mathscr{F}(\mathbf{h}_s)$, with $\mathbf{h}_s$ being the vector with elements $h_{s_i} = \kappa(\hat{\mathbf{x}}_s, \mathbf{z}_{s_i})$, where $\hat{\mathbf{x}}_s$ is the scale model learned from frame $t-1$ and $\mathbf{z}_{s_i}$ is the sample extracted from a new frame.

4.3. Implementation. The total procedure of our approach is summarized in Algorithm 1. In our approach, we use the Gaussian kernel function $\kappa(\mathbf{x}, \mathbf{x}') = \exp(-|\mathbf{x} - \mathbf{x}'|^2/\sigma^2)$ in the translation and scale detection; σ is the standard deviation. In tracking-by-detection method, the closer the samples to the currently tracked target center, the larger the probability the samples are the positive samples. Since the square loss of RLS with kernels allows for continuous values, we do not need to limit ourselves to binary labels. The line between classification and regression is essentially blurred. For the continuous training output, we choose the Gaussian function, which is known to minimize ringing in the Fourier domain [33]. Therefore, the desired outputs $\mathbf{y}$ and $\mathbf{y}_s$ both use the Gaussian functions that are expressed in

$$\begin{aligned} \mathbf{y} &= \exp\left(-\left(\frac{\mathbf{p} - \mathbf{p}^*}{\sigma} \right)^2 \right), \\ \mathbf{y}_s &= \exp\left(-\left(\frac{\mathbf{s} - \mathbf{s}^*}{\sigma_s} \right)^2 \right), \end{aligned} \tag{17}$$

where $\mathbf{p}$ represents a target location, $\mathbf{p}^*$ represents the coordinate of the tracked target center, $\mathbf{s}$ is a target scale with elements s_i ($1 \le s_i \le S$, where s_i is an integer), $\mathbf{s}^*$ is the centre scale of the target, and σ and σ_s are the standard deviations.

5. Experimental Results

To verify the efficiency of the method introduced above, we test the proposed tracking algorithm on 12 challenging video sequences which are from [29]. They have been widely used in many recent tracking papers and are summarized

Input: The ith frame video sequence V^i, Initial target position $\mathbf{p}^0$ and scale $\mathbf{s}^0$.
Output: Detected target position $\mathbf{p}^i$ and scale $\mathbf{s}^i$.
Repeat:
Crop out the searching region in frame i according to $\mathbf{p}^{i-1}$ and $\mathbf{s}^{i-1}$, and extract the sample $\mathbf{x}$;
//Position Detection:
(1) Compute the response $\hat{\mathbf{y}}$ with $\mathbf{x}$, $\mathbf{D}^{i-1}$ and $\mathbf{x}^{i-1}$ using (12);
(2) Find the target position $\mathbf{p}^i$ which maximizes $\hat{\mathbf{y}}$.
//Scale Prediction:
(3) Extract a sample $\mathbf{x}_s$ from V^i at $\mathbf{p}^i$ and $\mathbf{s}^{i-1}$;
(4) Compute the response $\hat{\mathbf{y}}_s$ with $\mathbf{x}_s$, $\mathbf{D}_s^{i-1}$ and $\mathbf{x}_s^{i-1}$ using (16);
(5) Find the target scale $\mathbf{s}^i$ which maximizes $\hat{\mathbf{y}}_s$.
//Model Online Update:
(6) Extract samples $\mathbf{z}$ and $\mathbf{z}_s$ from V^i at $\mathbf{p}^i$ and $\mathbf{s}^i$;
(7) Update $\mathbf{D}^i$ using (13), (14) and update $\mathbf{x}^i$ using (9);
(8) Update $\mathbf{D}_s^i$ using (13), (14) and update $\mathbf{x}_s^i$ using (9).
Until the End of the Video Sequence

Algorithm 1: Proposed tracking algorithm.

Table 1: The tracking sequences used in our experiments.

Sequence	Frames	Main challenges
Car4	659	Scale, illumination, pose variation, and background clutter
CarScale	252	Scale variation and occlusion
Dog1	1350	Scale and pose variation
Girl	502	Scale, pose variation, in-plane rotation, and occlusion
Trellis	569	Scale, illumination, and pose variation
Singer1	351	Illumination and scale variation
David	462	Illumination, scale, and pose variation
Woman	597	Nonrigid deformation and occlusion
Tiger1	354	Abrupt motion, pose variation, and occlusion
Skating1	400	Illumination, pose variation, background clutter, and occlusion
CarDark	393	Illumination variation and background clutter
Faceocc1	886	Occlusion

in Table 1. We provide both quantitative and attribute-based comparisons with 5 state-of-the-art trackers. The tracking results for 12 video sequences using 6 tracking algorithms are shown in Supplementary Material available online at http://dx.doi.org/10.1155/2015/238971.

5.1. Experiment Environment and Parameters. All our experiments are performed using MATLAB 2010a on a 3.4 GHz Intel core i3-2130 PC with 2 GB RAM. For fair evaluation, all the parameters are fixed for all the video sequences in our experiments. For the target of size $m \times n$ and the scale filter of size $S \times 1$, the standard deviations are set to $\sigma = \sqrt{mn}$ and $\sigma_s = \sqrt{S}$. The standard deviation for the Gaussian kernel is 0.2. The learning rate α is 0.075. The regularization parameter λ is 0.01. The scale of the scale filter is set to $S = 31$ and the scale rate is set to $r = 1.1$.

5.2. Performance Evaluation. In order to evaluate the overall performance of the proposed method, three evaluation metrics are used, namely, centre location error (CLE), success rate (SR), and distance precision (DP). The CLE is defined as the average Euclidean distance between the manually labeled ground truths and the detected centre locations of the target. Then we use the average CLE over all the frames of a sequence to evaluate the overall performance for the sequence. SR is computed by (18). DP is defined as the relative number of frames in a sequence whose CLE is smaller than a fixed threshold. The threshold is set to 20 pixels in our experiment:

$$\text{score} = \frac{\text{area}\left(R_t \cap R_g\right)}{\text{area}\left(R_t \cup R_g\right)}, \qquad \text{SR} = \frac{sn}{n}, \tag{18}$$

where score is the overlap score, R_t is the tracked bounding box, R_g is the ground truth bounding box, area represents the region area, $\cap$ and $\cup$, respectively, represent the intersection and union of two regions, sn is the number that we use to count the successfully tracked frames whose overlap score is larger than 0.5, and n is the total frames of one sequence.

5.3. Comparison with Original Update Scheme. To show the effect of the changed update scheme on tracking, we compute the average CLE, average SR, and average DP over 12 sequences for the CSK, CSK with new update scheme, CSK with scale prediction, and our tracker which includes a new update scheme and a scale prediction at the same time. Table 2 shows the comparison results, and the best results are shown in bold. From the table, we can see that the new update scheme improves the performance of the tracker compared to the original update scheme. The CSK with new update

Table 2: Comparison with original update scheme.

Method	Average CLE (in pixels)	Average SR (%)	Average DP (%)
CSK	40.2	53.5	61.8
CSK + new update scheme	27.0	54.9	69.5
CSK + scale prediction	9.6	79.1	90.3
Ours	**8.1**	**82.6**	**92.9**

Table 3: Centre location error (in pixels).

Sequence	MOSSE	WMILT	CT	KCF	CSK	Ours
Car4	105.2	85.7	77.1	9.5	19.1	**4.4**
CarScale	70.0	70.4	**16.0**	16.1	90.5	19.5
Dog1	**4.2**	6.7	9.1	4.4	4.9	5.3
Girl	52.4	51.0	40.5	32.5	36.0	**17.2**
Trellis	77.5	43.0	44.8	8.2	19.0	**7.8**
Singer1	112.2	16.8	19.1	12.8	13.9	**4.5**
David	129.1	24.4	**9.0**	9.5	17.2	9.5
Woman	355.5	126.2	119.3	10.1	236.9	**10.0**
Tiger1	32.6	**9.5**	22.2	18.0	26.1	11.7
Skating1	20.8	8.4	152.8	**7.7**	10.1	**7.7**
CarDark	3.2	60.8	47.0	5.8	**2.6**	2.7
Faceocc1	6.1	31.8	29.6	43.7	**5.5**	6.7
Average CLE	72.4	44.6	48.9	14.9	40.2	**8.1**

scheme approach reduces the average CLE by 13.2 pixels and improves the performance by 1.4% in average SR and 7.7% in average DP compared to the CSK. Our tracker reduces the average CLE by 1.5 pixels and improves the tracking performance by 3.5% in average SR and by 2.6% in average DP compared to the CSK with scale prediction approach. Our tracker achieves the best performance in terms of the average CLE, average SR, and average DP.

5.4. Comparison with CSK Tracker. From Tables 3–5, we can see that our tracker reduces the average CLE from 40.2 pixels to 8.1 pixels and improves the performance by 29.1% in average SR and 31.1% in average DP compared to the CSK. Our approach outperforms the CSK in terms of the average CLE, average SR, and average DP.

In order to show clearly, we use the Girl sequence as an example to analyze. Figures 2 and 3, respectively, show the partial tracking results and the three evaluation metrics plots. Figure 2 shows the tracking results on Girl sequence with scale variation, pose variation, rotation, and partial occlusion. When the girl undergoes the rotation at frame #110, the CSK tracker begins to drift. When the target size becomes smaller at frame #156, our tracked box becomes smaller at the same time and our tracker can track the girl accurately. The tracking error of the CSK tracker is accumulated as the target appearance varies. CSK has a great drift at frame #436 and fails to track the girl at frame #472. However, our tracker can track the girl successfully all the time. Figure 3 also shows that our approach is better than CSK.

Table 4: Success rate (%).

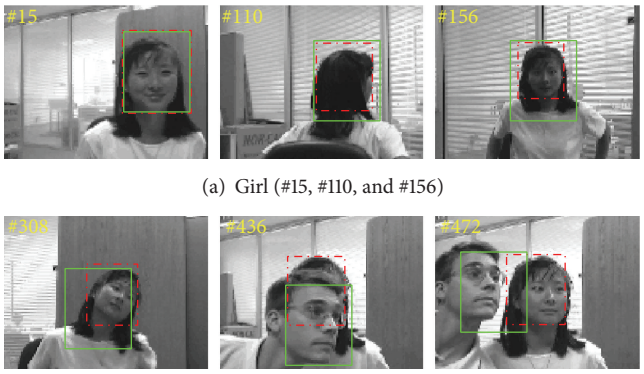
Sequence	MOSSE	WMILT	CT	KCF	CSK	Ours
Car4	27.5	24.6	27.6	36.7	27.6	**100**
CarScale	44.8	44.8	44.8	44.4	44.8	**81.0**
Dog1	65.3	62.7	59.4	65.3	65.1	**99.9**
Girl	35.6	27.7	30.7	**84.2**	51.5	76.2
Trellis	30.6	33.6	26.0	84.0	55.0	**89.8**
Singer1	29.6	27.6	27.6	29.6	29.6	**100**
David	14.0	51.6	**100**	**100**	57.0	25.8
Woman	23.6	16.2	15.4	**93.6**	23.8	86.1
Tiger1	39.4	**70.4**	56.3	69.0	50.7	57.7
Skating1	36.2	34.0	9.0	36.2	37.5	**75.5**
CarDark	96.4	0.3	12.2	72.3	**100**	**100**
Faceocc1	**100**	58.4	74.7	65.7	99.4	99.4
Average SR	45.3	37.7	40.3	65.1	53.5	**82.6**

Table 5: Distance precision (%).

Sequence	MOSSE	WMILT	CT	KCF	CSK	Ours
Car4	28.1	24.1	35.4	95.3	35.5	**100**
CarScale	65.1	63.1	65.1	**80.6**	65.5	76.2
Dog1	**100**	94.3	94.9	**100**	99.9	99.6
Girl	34.7	21.8	24.8	60.4	39.6	**65.3**
Trellis	34.1	45.0	25.3	**100**	75.6	**100**
Singer1	84.9	63.5	32.2	81.5	67.5	**100**
David	14.0	40.9	**100**	**100**	50.5	97.8
Woman	24.8	20.6	20.4	**93.8**	25.0	**93.8**
Tiger1	52.1	**93.0**	67.6	73.2	63.4	87.3
Skating1	70.0	97.8	11.7	**100**	87.0	96.0
CarDark	**100**	10.4	21.4	**100**	**100**	**100**
Faceocc1	**99.4**	19.1	32.0	64.6	**99.4**	98.9
Average DP	58.9	49.5	44.2	87.5	61.8	**92.9**

5.5. Comparison with State-of-the-Art Trackers. Since it is impractical to use all the existing tracking algorithms to validate the efficacy of our tracker, we compare the proposed algorithm with 5 state-of-the-art trackers: MOSSE tracker [22], Compressive Tracker (CT) [17], Weighted Multiple Instance Learning Tracker (WMILT) [34], KCF tracker with HOG features [28], and CSK tracker [24]. In order to compare fairly, we use the same parameters as the authors suggested in their papers and only change the target location and size used in the first frame.

5.5.1. Quantitative Analysis. We compute the median CLE, SR, and DP to evaluate the performance of 6 tracking methods on the 12 challenging video sequences in our experiments. The results are shown in Tables 3–5. The best results are reported in bold. The three tables show the quantitative results in which our tracker achieves the best or second best performance in most sequences in terms of CLE, SR,



(a) Girl (#15, #110, and #156)

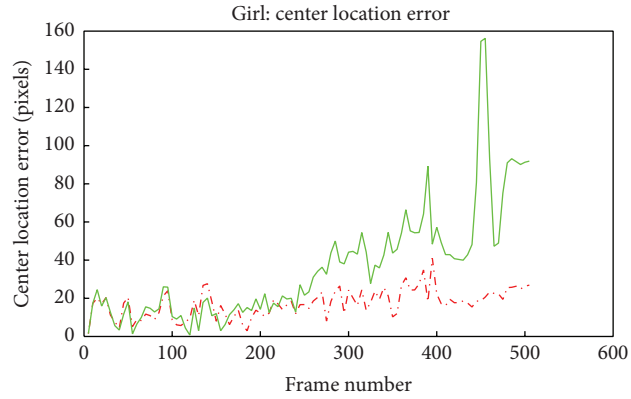
(b) Girl (#308, #436, and #472)

Figure 2: Partial tracking results compared to CSK: the plots of our tracker and CSK are, respectively, represented by red dot-dashed curve and green solid curve.

and DP. Compared to the second best tracker among the 6 trackers, our tracker reduces the average CLE by 6.8 pixels and improves the performance by 17.5% in the average SR and by 5.4% in the average DP. To describe the tracking results in detail, we give the center location error plots, the overlap score plots, and the distance precision plots which are shown in Figures 4–6 over 12 sequences for these trackers. From the figures, we can see that our tracker maintains a smaller centre location error, a higher overlap score, and a higher distance precision in general. The above analysis implies that our approach performs more accurate and stable results than the other 5 trackers.

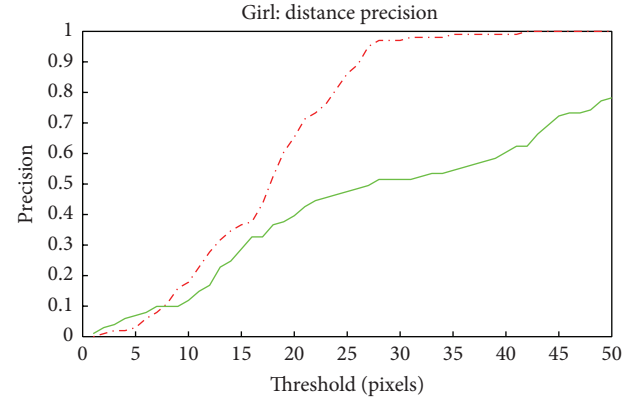
5.5.2. Qualitative Analysis

Scale, Illumination, and Pose Variation. Figures 7(a), 7(b), 7(c), and 7(d), respectively, illustrate the results on Car4, Singer1, Trellis, and David sequences with scale and illumination variations as well as pose changes. In Car4 sequence, the vehicle undergoes drastic illumination and scale changes especially when it passes beneath a bridge (see frame #230). Besides, the vehicle also undergoes background clutter. Only our approach and KCF are robust to these factors and perform well on this sequence. The HOG features are robust to illumination changes, but the background information in the tracked box of KCF accumulates because of the target scale variation, and KCF has a great drift at frame #641. However, our tracker can accurately detect the target position and scale all the time since it can predict the object scale in time. CT and WMILT use the discriminative classifiers learned by Harr-like features, MOSSE uses an adaptive correlation filter, and CSK brings kernelized correlation filters into tracking, but they perform poorly in this case. For the Singer1 sequence, the other trackers except our tracker fail to deal with the large scale, large illumination, and pose variation at the same time. Despite these challenges, our approach is able to track the target accurately. For the David indoor sequence shown in Figure 7(d), the person walks towards the moving camera, resulting in significant appearance variations due



(a) The center location error plot

(b) The overlap score plot



(c) The distance precision plot

Figure 3: Three evaluation metrics plots: the plots of our tracker and CSK are, respectively, represented by red dot-dashed curve and green solid curve.

to the illumination and scale change. CT, KCF, and our approach can successfully track the target in most frames of the David sequence. However, the target undergoes abrupt pose variation in the Trellis sequence, and only KCF and our tracker perform well. The CLE of our tracker is smaller and the SR of our tracker is higher.

Scale, Pose Variation, Occlusion, and Rotation. Figures 7(e) and 7(f), respectively, show the results on CarScale and Girl

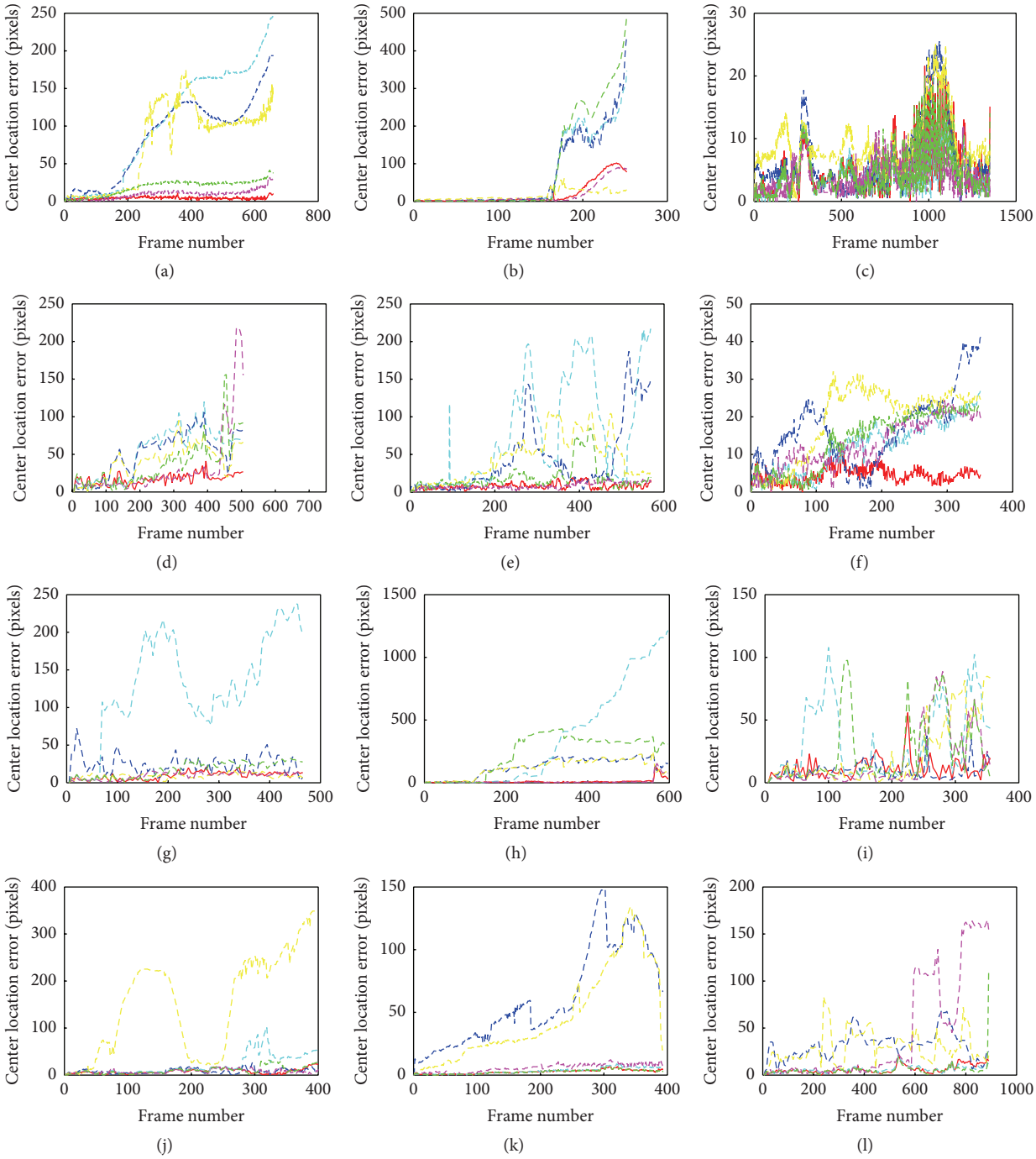


FIGURE 4: The centre location error plots: (a) Car4, (b) CarScale, (c) Dog1, (d) Girl, (e) Trellis, (f) Singer1, (g) David, (h) Woman, (i) Tiger1, (j) Skating1, (k) CarDark, and (l) Faceocc1. The plots of our tracker, MOSSE, WMILT, CT, KCF, and CSK are represented by red solid curve, cyan dashed curve, blue dashed curve, yellow dashed curve, magenta dashed curve, and green dashed curve.

sequences with scale variation and partial occlusion. The car moves from far to near and undergoes occlusion by trees in the CarScale sequence. Both KCF and our tracker can complete the total tracking task for the sequence, but the SR of our tracker is higher. In Figure 7(f), the girl also undergoes in-plane rotation and pose variation (see frames #141, #180) which make the tracking more difficult. Only our tracker is able to track the target successfully in most frames of this sequence.

Background Clutter, Illumination, Pose Variation, and Occlusion. The targets in the Skating1 and CarDark sequences undergo background clutter, illumination, and pose changes. For the Skating1 sequence in Figure 7(g), the target also

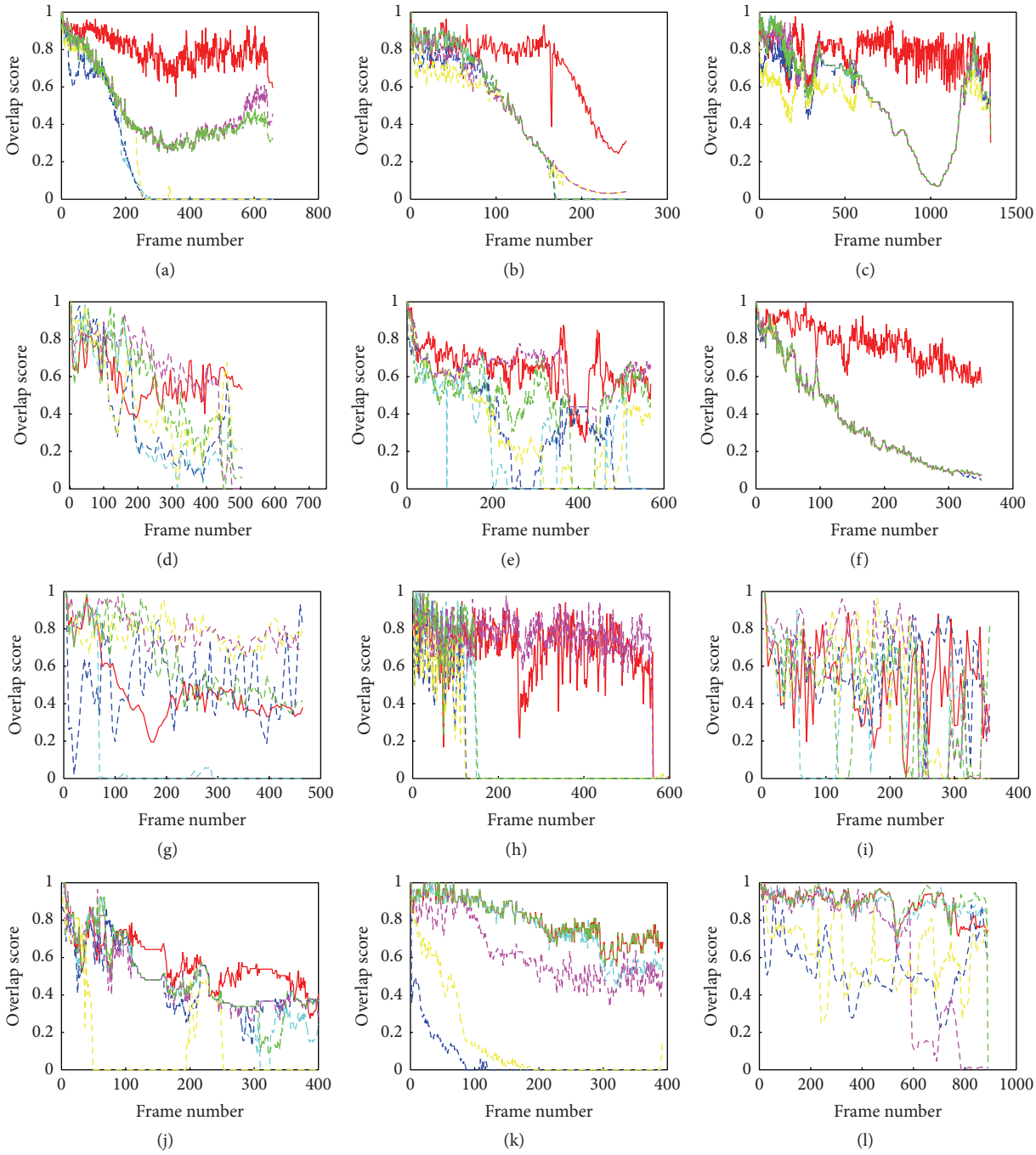


FIGURE 5: The overlap score plots: (a) Car4, (b) CarScale, (c) Dog1, (d) Girl, (e) Trellis, (f) Singer1, (g) David, (h) Woman, (i) Tiger1, (j) Skating1, (k) CarDark, and (l) Faceocc1. The plots of our tracker, MOSSE, WMILT, CT, KCF, and CSK are represented by red solid curve, cyan dashed curve, blue dashed curve, yellow dashed curve, magenta dashed curve, and green dashed curve.

undergoes partial occlusion (see frame #163). Only KCF and our tracker perform well during the tracking process, but our approach performs better in terms of CLE and SR. For the CarDark sequence in Figure 7(h), MOSSE, CSK, and our tracker provide promising results compared to other trackers.

Scale, Pose Variation, Occlusion, and Abrupt Motion. Figure 7(i) shows the Dog1 sequence with scale and pose variation. MOSSE and KCF as well as our approach perform well on the sequence. In addition, our tracker achieves the best performance in terms of SR. For the Tiger1 sequence as shown in Figure 7(j), the object undergoes abrupt motion, pose variation, and partial occlusion. Only WMILT and our tracker can adapt to these factors. The partial occlusion occurs in the Woman and Faceocc1 sequences (Figures 7(k) and 7(l)) at times. The Woman sequence has nonrigid

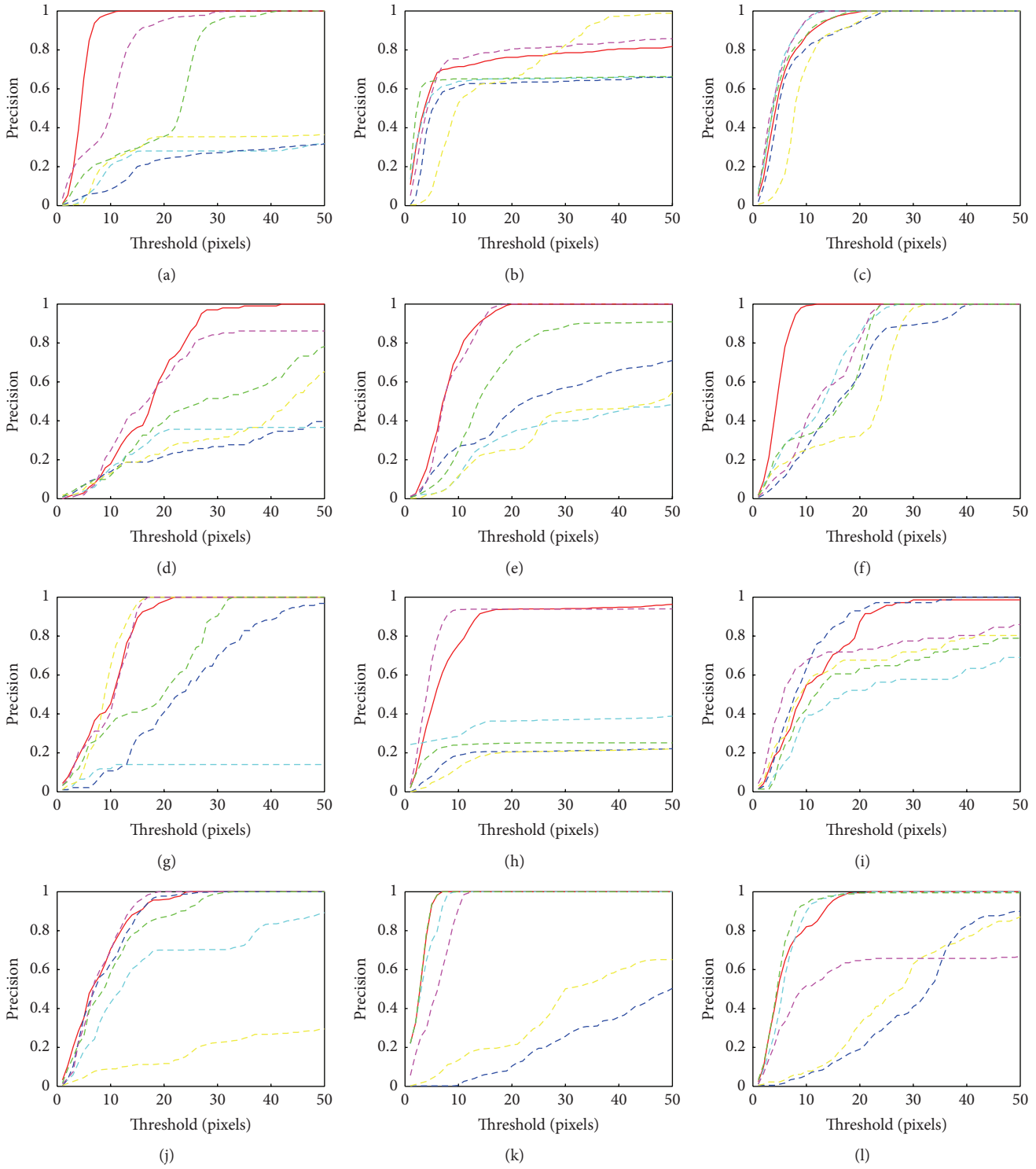
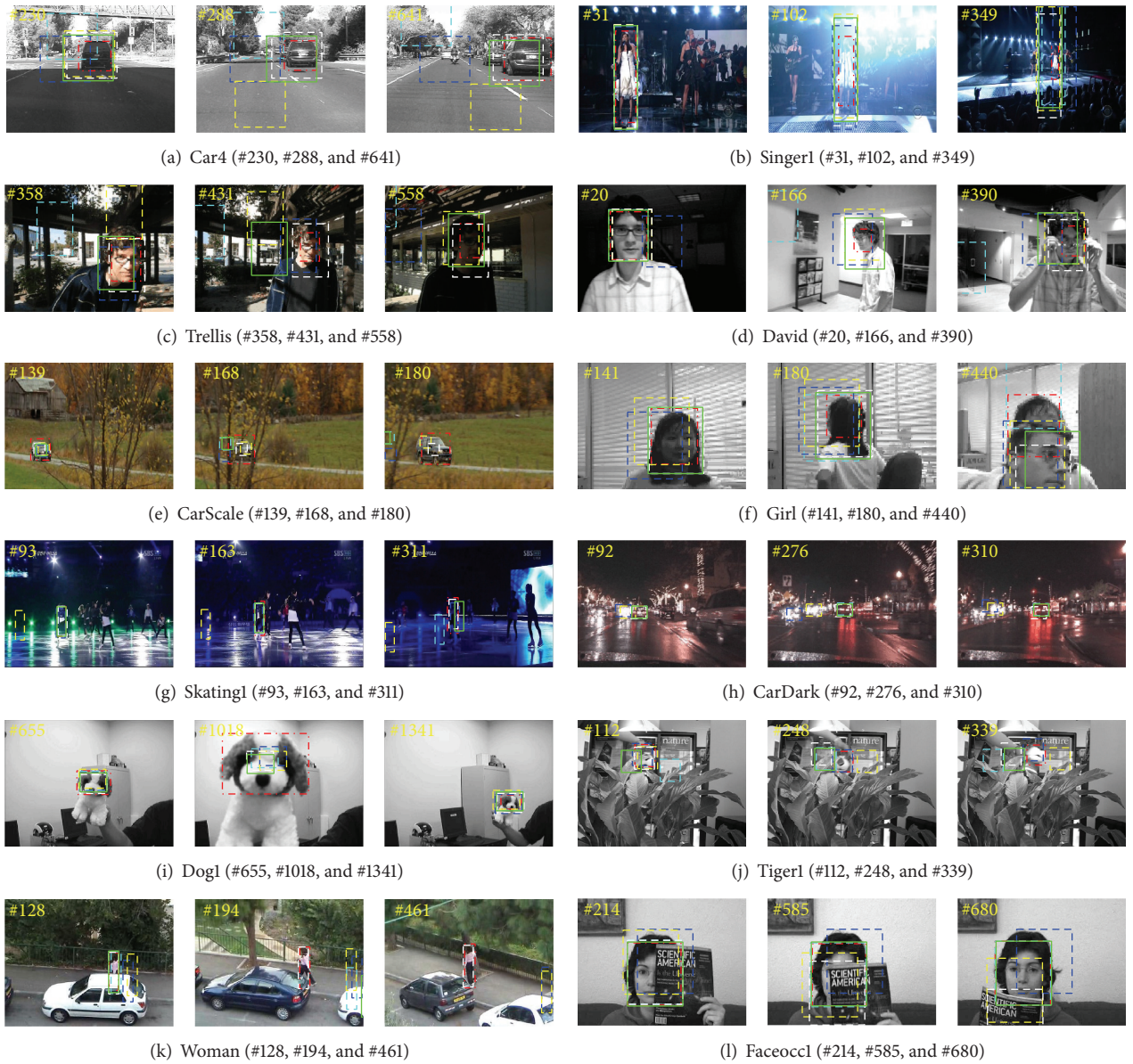


Figure 6: The distance precision plots: (a) Car4, (b) CarScale, (c) Dog1, (d) Girl, (e) Trellis, (f) Singer1, (g) David, (h) Woman, (i) Tiger1, (j) Skating1, (k) CarDark, and (l) Faceocc1. The plots of our tracker, MOSSE, WMILT, CT, KCF, and CSK are represented by red solid curve, cyan dashed curve, blue dashed curve, yellow dashed curve, magenta dashed curve, and green dashed curve (the chosen threshold is 50 pixels).

deformation and heavy occlusion at the same time. All the other trackers fail to successfully track the object except KCF and our tracker. But, in the Faceocc1 sequence, only MOSSE, CSK, and our approach perform well.

5.6. Discussion. From the above qualitative and quantitative analyses, our tracker outperforms other trackers in most cases. The reason is that our tracker not only can predict the target location, but also is able to estimate the target



(a) Car4 (#230, #288, and #641)

(b) Singer1 (#31, #102, and #349)

(c) Trellis (#358, #431, and #558)

(d) David (#20, #166, and #390)

(e) CarScale (#139, #168, and #180)

(f) Girl (#141, #180, and #440)

(g) Skating1 (#93, #163, and #311)

(h) CarDark (#92, #276, and #310)

(i) Dog1 (#655, #1018, and #1341)

(j) Tiger1 (#112, #248, and #339)

(k) Woman (#128, #194, and #461)

(l) Faceocc1 (#214, #585, and #680)

Figure 7: Partial tracking results: the plots of our tracker, MOSSE, WMILT, CT, KCF, and CSK are represented by red dash-dot box, cyan dashed box, blue dashed box, yellow dashed box, white dashed box, and green solid box.

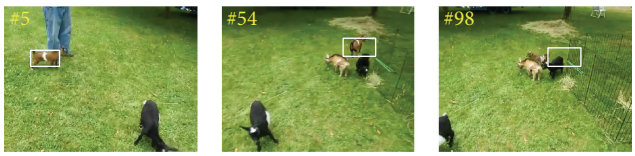
scale accurately at the same time. As to the computational complexity, the most time-consuming part of our tracker is to compute the latent HOG feature vectors of all the candidate samples. Our tracker is implemented in MATLAB, which runs at about 15 frames per second (FPS) on an Intel core i3-2130 3.4 GHz CPU with 2 GB RAM. Our tracker performs well in the above experiments, but drifts are also observed when the initial target is very little (e.g., see Freeman3 and Freeman4 sequences) and when the target moves unstably all the time (e.g., see Goat sequence as shown in Figure 8(c)). Figures 8(a) and 8(b), respectively, show the tracking results of our tracker over the Freeman3 and Freeman4 sequences, where the initial targets are very little (12×13 pixels in Figure 8(a), 15×16 pixels in Figure 8(b)). Our tracker can not estimate the increasing scale in the two sequences. This is because the HOG features perform poorly at low resolutions. In Figure 8(c), the goat moves unstably all the time (see frames #5, #54, and #98). Our tracker drifts away because of the accumulated online updated error from the continuous unstable motion.

6. Conclusion

Based on the framework of tracking with kernelized correlation filter and tracking-by-detection method, we develop a robust visual correlation tracking algorithm with improved tracking performance in this paper. Our tracker estimates the target translation and scale variations effectively and efficiently by learning the kernelized correlation filters. By accurately estimating the target scale in the tracking,

(a) Freeman3 (#4, #410, and #440)

(b) Freeman4 (#10, #202, and #258)



(c) Goat (#5, #54, and #98)

Figure 8: Three failed tracking cases.

our tracker can obtain more useful information from the target and reduce the interference from background. The translation is estimated by modeling the temporal context correlation and the scale is estimated by searching the tracked target appearance pyramid. In addition, we further develop an update scheme that takes all the previous frames into consideration when computing the current model. Experimental results on challenging sequences clearly show that our approach outperforms state-of-the-art tracking algorithms in terms of efficiency, accuracy, and robustness.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors truly thank the reviewers for valuable advice and comments. This work is supported by the National High Technology Research and Development Program of China (Grant no. 2014AA7031010B).

References

[1] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1090–1097, IEEE, Columbus, Ohio, USA, June 2014.

[2] J. Fang, Q. Wang, and Y. Yuan, "Part-based online tracking with geometry constraint and attention selection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 854–864, 2014.

[3] X. Q. Zhang, W. M. Hu, S. Y. Chen, and S. Maybank, "Graph-embedding-based learning for robust object tracking," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 1072–1084, 2014.

[4] E. Chen, O. Haik, and Y. Yitzhaky, "Detecting and tracking moving objects in long-distance imaging through turbulent medium," *Applied Optics*, no. 6, pp. 1181–1190, 2014.

[5] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1910–1917, Providence, RI, USA, June 2012.

[6] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2042–2049, IEEE, Providence, RI, USA, June 2012.

[7] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.

[8] J. M. Danell, G. Hager, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference (BMVC '14)*, Nottingham, UK, September 2014.

[9] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the 17th British Machine Vision Conference*, pp. 47–56, September 2006.

[10] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[11] S. Wang, H.-C. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1323–1330, IEEE, Barcelona, Spain, November 2011.

[12] K. H. Zhang, L. Zhang, and M. H. Yang, "Fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2002–2015, 2014.

[13] F. Yang, H. C. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1639–1651, 2014.

[14] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[15] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 49–56, San Diego, Calif, USA, June 2010.

[16] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 263–270, IEEE, Barcelona, Spain, November 2011.

[17] K. H. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 864–877, 2012.

[18] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1910–1917, Providence, RI, USA, June 2012.

[19] Q. Tu, Y. P. Xu, and M. L. Zhou, "Robust vehicle tracking based on Scale Invariant Feature Transform," in *Proceedings of*

the IEEE International Conference on Information and Automation (ICIA '08), pp. 86–90, Changsha, China, June 2008.

[20] M. Jiang, L. Zhang, and Y. L. Huang, "Object tracking based on particle filter and scale invariant feature transform," in *Proceedings of the IEEE International Conference on Multimedia Technology (ICMT '10)*, pp. 1–4, IEEE, Ningbo, China, October 2010.

[21] L. Wei, X. Xudong, W. Jianhua, Z. Yi, and H. Jianming, "A SIFT-based mean shift algorithm for moving vehicle tracking," in *Proceedings of the 25th IEEE Intelligent Vehicles Symposium (IV '14)*, pp. 762–767, Dearborn, Mich, USA, June 2014.

[22] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2544–2550, San Francisco, Calif, USA, June 2010.

[23] B. V. K. V. Kumar, A. Mahalanobis, and R. D. Juday, *Correlation Pattern Recognition*, Cambridge University Press, 2005.

[24] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the European Conference on Computer Vision*, pp. 702–715, Florence, Italy, October 2012.

[25] J. F. Henriques, J. Carreira, R. Caseiro, and J. Batista, "Beyond hard negative mining: efficient detector learning via block-circulant decomposition," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 2760–2767, Sydney, Australia, December 2013.

[26] H. K. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '13)*, pp. 3072–3079, IEEE, Sydney, Australia, December 2013.

[27] V. N. Boddeti, T. Kanade, and B. V. K. V. Kumar, "Correlation filters for object alignment," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2291–2298, Portland, Ore, USA, June 2013.

[28] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[29] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: a benchmark," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2411–2418, Portland, Ore, USA, June 2013.

[30] R. M. Gray, *Toeplitz and Circulant Matrices: A Review*, Now Publishers, Norwell, Mass, USA, 2006.

[31] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least-squares classification," in *Nato Science Series Sub Series III Computer and Systems Sciences*, vol. 190, pp. 131–154, IOS Press, Amsterdam, The Netherlands, 2003.

[32] B. Schőlkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, 2002.

[33] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2008.

[34] K. H. Zhang and H. H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognition*, vol. 46, no. 1, pp. 397–411, 2013.