Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 5986062, 14 pages
https://doi.org/10.1155/2018/5986062

Research Article

A Fast Object Tracker Based on Integrated Multiple Features and Dynamic Learning Rate

Jianming Zhang[1,2] **You Wu**[1,2] **Xiaokang Jin**[1,2] **Feng Li**[1,2] **and Jin Wang**[1,2]

[1] *Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha 410114, China*
[2] *School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China*

Correspondence should be addressed to Jin Wang; jinwang@csust.edu.cn

Received 10 August 2018; Revised 23 November 2018; Accepted 6 December 2018; Published 24 December 2018

Academic Editor: George A. Papakostas

Object tracking is a vital topic in computer vision. Although tracking algorithms have gained great development in recent years, its robustness and accuracy still need to be improved. In this paper, to overcome single feature with poor representation ability in a complex image sequence, we put forward a multifeature integration framework, including the gray features, Histogram of Gradient (HOG), color-naming (CN), and Illumination Invariant Features (IIF), which effectively improve the robustness of object tracking. In addition, we propose a model updating strategy and introduce a skewness to measure the confidence degree of tracking result. Unlike previous tracking algorithms, we judge the relationship of skewness values between two adjacent frames to decide the updating of target appearance model to use a dynamic learning rate. This way makes our tracker further improve the robustness of tracking and effectively prevents the target drifting caused by occlusion and deformation. Extensive experiments on large-scale benchmark containing 50 image sequences show that our tracker is better than most existing excellent trackers in tracking performance and can run at average speed over 43 fps.

1. Introduction

It is difficult to accurately estimate the location of target in a video due to the complex causes such as occlusion, deformation, illumination variation, background clutter, and scale variations, all of which have brought difficulties to tracking. Although object tracking has been successfully used in robotics, video surveillance, human-computer interaction, automation, etc., we still need to find an effective and robust tracker.

Most of existing tracking methods mainly include two categories: one is generative method and the other is discriminative method. Generative trackers firstly construct a target appearance model, then match it with the candidate target regions, and take the candidate region with the highest similarity to the target region as the tracking result. There are many generative algorithms such as sparse representation [1–3], density estimation [4, 5], and incremental subspace learning [6]. In contrast, discriminative trackers use sample data to learn a binary classifier which can discriminate tracked target from its background areas. Discriminative trackers include multiple instance learning (MIL) [7], compressive tracking (CT) [8], tracking-learning-detection (TLD) [9], support vector machines (SVMs) [10–12], and online adaboost (OAB) [13, 14]. MIL tracker employs a set of generalized Haar-like features to represent the image patch and each feature consisting of two to four rectangles. MIL also trains a classifier with multiple instances learning to achieve superior results. CT tracker based on compressed sensing enhances tracking efficiency thanks to Haar-like features are reduced by random measurement matrix conforming to the restricted isometry property (RIP), and a simple naive Bayes classifier is used to classify the features after dimensionality reduction. In recent years, the discriminative trackers based on correlation filters have raised much attention in the field of visual tracking due to their outstanding performance in computing efficiency. Bolme et al. [15] firstly introduce correlation filters into visual tracking and learn filter by minimizing the output sum of squared error on grayscale images. Henriques et al. [16] figure out a circulant structure with kernel (CSK) method achieving

an amazing speed on tracking benchmark [17], but CSK only uses gray features which are less effective in representing the target appearance model. Later, the performance of tracking has been further improved in kernelized correlation filters (KCF) [18] tracker. Discriminative scale space tracking (DSST) [19] enhances the tracking accuracy by multichannel HOG features instead of the low dimensional gray features. Danelljan et al. [20] exploit color attributes (CN) for tracking and extend the input color features from single channel to multiple channels.

However, all above-mentioned trackers only use single feature which limits the power of target representations when the object appearances undergo challenges such as occlusion and illumination changes. As a result, the ideal tracking results are often difficult to obtain. To overcome the limitation of single feature on target tracking, scale adaptive multiple features (SAMF) [21] tracker integrates HOG features and CN features based on correlation filter to improve tracking accuracy. Lan et al. propose a discriminative feature learning method in [22, 23], which can exploit the representation and discriminative abilities of multiple features by separating out contaminated features. Sum of template and pixel-wise learner (Staple) [24] tracker combines the response maps of the HOG template and global color histogram both of which are learned independently in previous estimated translation to enhance tracking performance. Convolutional neural network (CNN) has found a broad application in pattern classification [25] and text processing [26] because of its powerful feature representation ability. Several existing tracking approaches based on CNN such as a deep compact image representation for tracking (DLT) [27], hierarchical convolutional features for tracking (HCF) [28], hedged deep tracking (HDT) [29], and spatial and semantic convolutional features for tracking (DSCF) [30] have been proposed. They all extract rich features from CNN to precisely predict the target position and have shown excellent performance. Although these algorithms based on features fusion or CNN features are satisfactory in constrained environment, these methods do not address the vital problem with respect to the model update mechanism with a constant learning rate which are prone to drifting in tracking due to inaccurate prediction. For the drifting problem, thus, TLD tracker [9] combines tracking learning with detection, the mechanism performs well in presence of occlusion, deformation. The long-term correlation tracking (LCT) [31] can prevent significant occlusion by using an online detector to detect the target again when wrong tracking results appear. SUN et al. [32] present mixed classifier decision compressive tracking (MDCT) method to locate the target and update the models by using different learning rates to improve the tracking accuracy.

In this paper, to overcome the problem that DSST tracker cannot describe target well and its model updating strategy which uses constant learning rate is unable to update filters adaptively, we propose a fast object tracker based on integrated multiple features and dynamic learning rate. We integrate gray features, HOG, CN, and IIF [33] to improve the target description ability of algorithm while preserving the performance of tracking under complex circumstances. Meanwhile, for the problem of constant learning rate, we apply the criteria of skewness to our approach. Skewness can reflect the confidence degree of the tracking results via fluctuation of response map. By comparing the skewness values between two adjacent frames, our approach can adaptively choose a learning rate to update the model in tracking. To validate the contribution of our approach, we perform the extensive experiments on a popular benchmark dataset [17] with 50 image sequences and compare our proposed approach with 12 excellent algorithms using precision and success rate. Experimental results show that our tracker performs significantly against existing trackers in the aspect of accuracy and robustness of tracking, while maintaining a high average speed which exceeds 40 frames per second.

The organizational structure of the paper is shown below. We first introduce DSST tracker in Section 2 and then describe our approach in Section 3. Section 4 demonstrates the experimental results on benchmark dataset. Conclusions are finally given in Section 5.

2. The DSST Tracker

DSST tracker [19] has obtained impressive results on tracking benchmark and has some significant ideas relevant to our work. The algorithm separately learns correlation filters for translation and scale estimation. For the translation estimation, the DSST tracker trains an optimal correlation filter relying on the high-dimensional HOG features and then employs the filter to determine target location of next frame. Equipped with the estimated translation, the multiscale filters which use HOG features are applied to obtain accurate target size. We briefly describe the main ideas of DSST tracker in the following.

2.1. Translation Estimation. In DSST tracker, we crop an image patch $f \in \mathbb{R}^{M\times N}$ where target is located and extract d-dimensional feature map from the image patch to train translation filters. Considering the multidimensional feature maps of image patch, we let f^l denote the l-th dimension feature map of f, $l \in \{1, \ldots, d\}$. Per feature dimension has a corresponding filter h^l. The l-th feature dimension has a single filter h^l, and all of these h^l can be concatenated into optimal correlation filter h which obtained by minimizing the cost function:

$$\varepsilon = \left\| \sum_{l=1}^{d} h^l * f^l - g \right\|^2 + \lambda \sum_{l=1}^{d} \left\| h^l \right\|^2 \tag{1}$$

where g represents 2-dimensional Gaussian function in which its peak at the target center of the image patch f. $\lambda\,(\lambda > 0)$ denotes a regularization parameter, and $*$ is circular correlation. Note that the minimization issue in (1) can be solved by transforming (1) to the Fourier domain using Parseval's formula. The solution to (1) can be availably obtained by

$$H^l = \frac{\overline{G} F^l}{\sum_{k=1}^{d} \overline{F^k} F^k + \lambda} \tag{2}$$

where G and F^l denote the Discrete Fourier Transform (DFT) of g and f^l, respectively, and the bar $\overline{(\cdot)}$ indicates complex conjugation.

In (2), we only compute the correlation filter of a training sample. In practice, we find an optimal filter by minimizing the output error over all training patches, but it will lead to complex computations when requiring solving a $d \times d$ linear system of equations. In order to obtain high computational efficiency, A^l_{t-1} and B_{t-1} are defined as the numerator and denominator of filter H^l_{t-1} in the (t-1)-th frame, respectively. In the t-th frame, the numerator A^l_t and denominator B_t of H^l_t in (2) are updated separately in the following iterative ways:

$$A^l_t = (1-\eta)\, A^l_{t-1} + \eta \overline{G_t} F^l_t \tag{3}$$

$$B_t = (1-\eta)\, B_{t-1} + \eta \sum_{k=1}^{d} \overline{F^k_t} F^k_t \tag{4}$$

where η is the learning rate. Given an image patch z cropped from a new frame, d-dimensional feature maps are extracted from z. The correlation scores y can be computed by

$$y = \mathscr{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{A^l} Z^l}{B + \lambda} \right\} \tag{5}$$

where $\mathscr{F}^{-1}$ denotes the inverse DFT operator. Z^l denotes the Discrete Fourier Transform (DFT) of z^l. The new target position is found via the maximal response value of y.

2.2. Scale Estimation. In the actual tracking, the scale of target often changes because of the complexity of tracking environment. In order to solve problem of changing target size, the DSST tracker proposes a novel approach to predict the target scale. After determining the position of target, we construct scale pyramid by multiscale sampling in target area. Let $P \times Q$ denote the target size of the t-th frame, for each $i \in \{\lfloor -(S-1)/2 \rfloor, \ldots, \lfloor (S-1)/2 \rfloor\}$, an image patch J_i with the size of $a^i P \times a^i Q$ centered at the estimated target location of the t-th frame is cropped. Here, a denotes the scale factor and S is the number of scales. The set of image patch J consisted of all these image patch J_i. We extract d-dimensional HOG features from image patch set J to train scale filters h_{scale}. Similar to translation estimation, (3) and (4) are used to update the scale filters h_{scale}, but the desired correlation output g is a 1-dimensional Gaussian function. We get the response scores between the scale filter and image patch J_i by (5); the optimal scale of target can be obtained by maximum response scores.

3. Our Approach

DSST tracker uses HOG features for tracking, which only reflects partial characteristic of target and is easy to affect the robustness of tracking. Moreover, DSST tracker updates the filters using a fixed learning rate. However, the target appearance is dynamically changing in the tracking, so the DSST tracker cannot ensure the target model is updated with a reasonable learning rate. Therefore, in this paper, we improved the DSST algorithm by feature integration and model updating strategy with dynamic learning rate. The flowchart of our algorithm is shown in Figure 1. Like DSST, our tracking task is composed of two parts: translation and scale estimation. However, our algorithm fuses gray features, HOG, CN, and IIF [33] for translation estimation. We transform the multichannel fusion features extracted from image patch into Fourier domain in current frame and then use (5) to get the maximum response in the location of new target. For scale estimation, our method uses the same procedure as DSST, which uses the HOG features to train scale filter. With respect to the model update for translation filter, we adopt a new model updating strategy with dynamic learning rate which helps our algorithm to achieve significant performance gain in object tracking. With respect to the model update for scale filter, we follow the method in DSST.

We introduce the integration of multiple features for our tracking in Section 3.1. The novel model updating strategy is investigated in Section 3.2.

3.1. The Integration of Multiple Features Based on DSST. Integration features can provide richer representation of target than single feature. In this paper, we integrate gray features, HOG, CN, and IIF together based on DSST tracker.

HOG features are commonly used by various algorithms in the field of computer vision and can well show the edge and gradient information of the target. It divides the image patch into small connected regions which are also called cells. The gradient direction or edge orientation histogram are collected on the pixels of each cell. Compared with other features, HOG has many advantages that it can maintain favorable invariability in geometric and illumination. CN, based on the color names in English linguistics, are assigned by 11 color labels which can represent color names in real word. RGB color image is mapped to color-naming space via the mapping methods in [34]. We incorporate it into our integration scheme because it is robust to scale variation and rotation. IIF are obtained by transforming image into CIE Lab color space and then perform a nonparametric local rank transformation [35] on the image brightness channel. IIF features can enhance the ability of tracker to suppress intense illumination changes. The gray features only contain brightness channel and are the simplest features. Figure 2 shows the visualized results of gray features, HOG, CN, and IIF, respectively, on the *Bolt* sequence.

The four types of features are complementary to each other. In our work, we extract one-channel gray features, 31-channel HOG features, ten-channel CN features, and one-channel IIF from the image patch, respectively. Totally 43-channel features represent the target appearance. Note that the sizes of the four features are different from each other and these feature sizes should be normalized to a fixed size. Afterwards, we concatenate these normalized features together, which significantly enhance the performance of our proposed tracker.

3.2. Model Updating Strategy. The conventional algorithms use a constant learning rate to update the model. However, the target appearance is constantly changing due to the

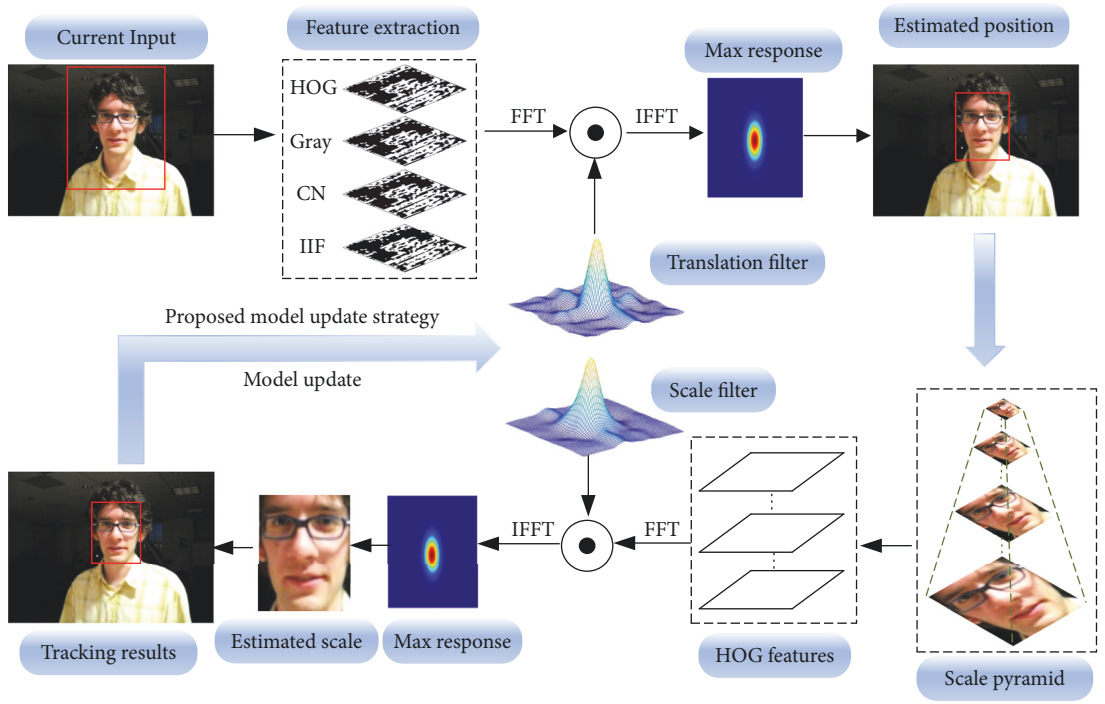
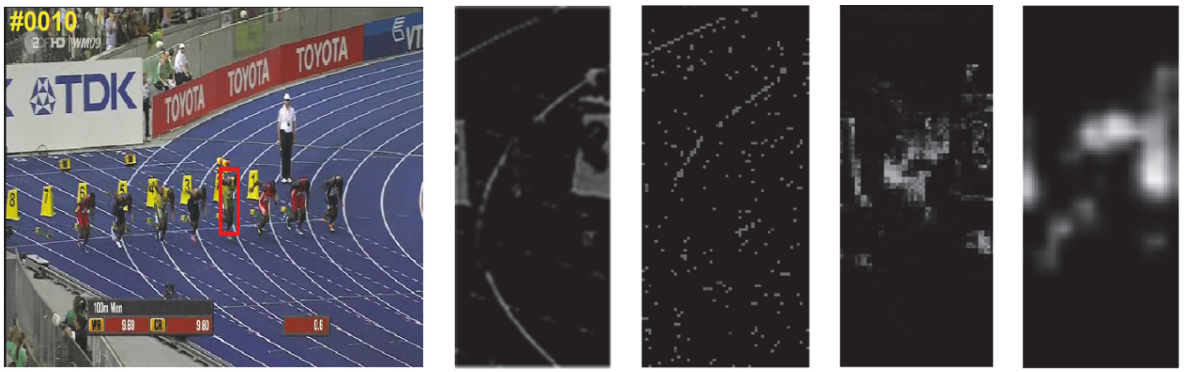


Figure 1: Framework of our proposed tracking algorithm.

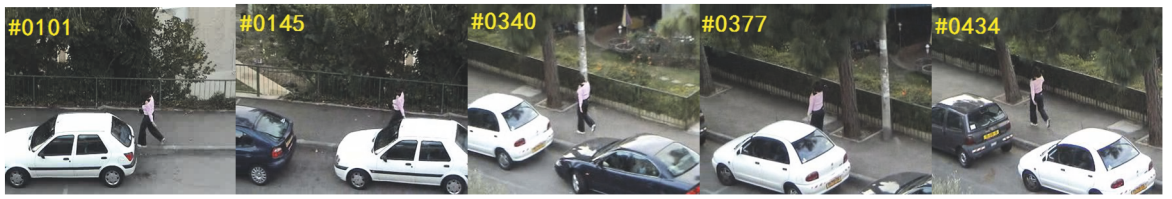


(a) Original image (b) Gray features (c) HOG (d) CN (e) IIF

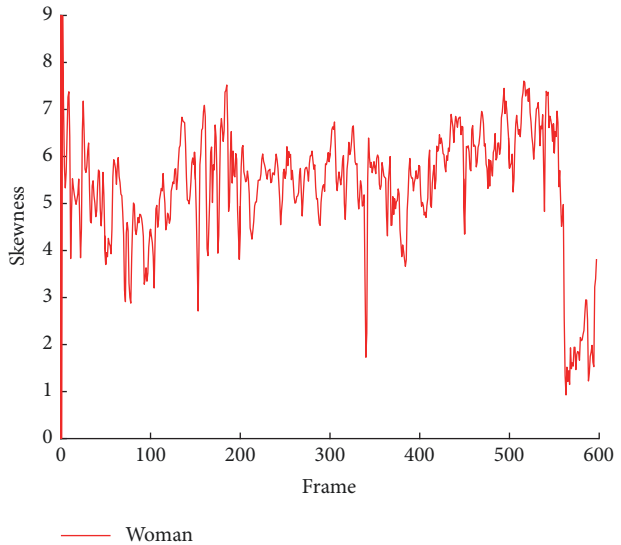
Figure 2: Visualizations for different features of *Bolt* sequence.

influence of deformation, occlusion, scale variations, and other factors in target tracking. Constant learning rate cannot cope with these interference factors effectively. The learning rate controls the updating degree of the target template. A higher learning rate can prevent insufficient updating of samples when the target appearance changes. But this will increase the probability of adding negative samples. In contrast, a lower learning rate can avoid learning more background information. Simultaneously, it is easy to suffer from target appearance deformation. Therefore, how to design a reasonable mechanism for dynamically updating learning rate is important.

In object tracking, the maximum response value in the response map is regarded as the target location, and other nontarget responses are generally much smaller than the maximum response value. However, in practice, the target is often disturbed by many factors, such as complex background, occlusion, and illumination variation, which may lead to some nontarget responses being closer to the target response value. As a result, the tracked target may be difficult to distinguish. The fluctuation degree of response map can reflect the quality of tracking results to a certain extent. In our method, in order to measure the fluctuation of the response map, we introduce a new criterion called skewness. It is a measure of the deviation direction and degree of data distribution and can reflect the asymmetric degree of data distribution. The relationship between skewness of two sequential frames can be used to decide the necessity of learning rate updates. The skewness value of response map in the t-th frame is defined as



(a) The challenging situations of occlusion on the *Woman* sequence



(b) The distribution of skewness on *Woman* sequence

FIGURE 3: The analysis of skewness on *Woman* sequence.

$$SK_t = \frac{(1/MN)\sum_{n-1}^{N}\sum_{m-1}^{M}\left(E\left(m,n\right)-\mu_t\right)^3}{\left[(1/MN)\sum_{n-1}^{N}\sum_{m-1}^{M}\left(E\left(m,n\right)-\mu_t\right)^2\right]^{3/2}} \quad (6)$$

where E is the response map obtained by (5) in the t-th frame and μ_t is its mean. M denotes the width of response map and N denotes the height of response map. The larger the skewness is, the greater the response value of the target is than the nontarget response; thus the tracking result in current frame is more reliable. On the contrary, it indicates that the difference between response value of target and nontarget is not significant when the skewness becomes smaller, and the tracking result in the current frame is disturbed. In these cases, we should choose appropriate learning rate to update target appearance.

Figure 3(b) shows the skewness of each frame on the *Woman* sequence. It can be seen from Figure 3(a), the target is partly occluded by the car in the 145th frame and interfered by a lamppost in the 340th frame. Consequently, the values of skewness are affected and reduced to lower points in Figure 3(b). When the occlusion disappears in the 434th frame, the value of skewness recovers to a higher point.

In our approach, we use the following three ways to update the learning rate:

$$\eta_o \longleftarrow \eta \quad (7)$$

$$\eta_d \longleftarrow \eta - \eta\frac{1}{n_1+1} \quad (8)$$

$$\eta_i \longleftarrow \eta + (1-\eta)\frac{1}{n_2+1} \quad (9)$$

where $\eta \in (0,1)$ is an original learning rate which is defined as η_o in this paper. η_d and η_i are new learning rates. $\eta_d \in (0,\eta)$ gradually decreases with the increase of n_1. $\eta_i \in (\eta,1)$ gradually become larger with the increase of n_2. There are more details in Remarks 1 and 2. n_1 denotes the number of frames when the skewness of the (t-1)-th frame minus the skewness of the t-th frames over θ. Similarly, n_2 denotes the number of frames when the present skewness minus previous is greater than θ. We define the threshold θ ($\theta > 0$) as the skewness difference of two adjacent frames. Whether a new learning rate is applied to the t-th frame depends on the difference of the two skewness values between two adjacent frames. A more intuitive updating strategy for learning rate is displayed in Table 1. If the value of skewness satisfies condition 1, it shows that the confidence level of tracking results in the t-th frame is not as reliable as the previous frame and we should reduce the learning rate to adapt the quick changes in the appearance of target. If condition 2 is satisfied, it indicates that the tracking results are reliable in the t-th frame and we should increase the learning rate to adapt the rapid changes of target appearance. There are slow changes in the appearance of target when condition 3 is satisfied.

Table 1: The updating strategy of learning rate.

No	State of learning rate	Value of learning rate	Condition in the t-th frame
1	Update	η_d	$SK_{t-1} - SK_t > \theta$
2	Update	η_i	$SK_t - SK_{t-1} > \theta$
3	Not update	η_o	$\left\|SK_t - SK_{t-1}\right\| < \theta$

Input: Initial target position p_1 and scale s_1
Output: Estimated target position p_t and scale s_t
For t=2: n
Translation estimation
1: Crop out the translation sample x_{trans} from the input image at the previous target position and extract the four types of features.
2: Compute the translation correlation filters h_{trans} using Eq. (2).
3: Estimate the new position p_t through the maximum response of Eq. (5).
Scale estimation
4: Construct the scale pyramid centered at the estimated position p_t.
5: Compute the scale correlation filter h_{scale} using Eq. (2).
6: Estimate the optimal scale s_t through the maximum response of Eq. (5).
Model update
7: Calculate SK_t with Eq. (6), update the learning rate according to relation of skewness between two adjacent frames.
8: Update the translation filter using new learning rate in Eqs. (3) and (4).
9: Update the scale filter using the original learning rate in Eqs. (3) and (4).
END

Algorithm 1: Overall procedure of our algorithm.

In this case, we apply the initial learning rate $\eta_o$to the t-th frame.

Remark 1. With the iteration of (8), η_d is getting smaller and can be used as a new learning rate whose value requires to be in the interval (0,1).

Proof. Given original learning rate $\eta \in (0, 1)$, the corresponding number of frames $n_1 \in N^+$ meets condition 1 in Table 1. According to (8), a new learning rate η_d is updated as $\eta - \eta(1/(n_1 + 1))$. Therefore, we have two inequalities as follows:

$$\eta_d = \eta - \eta\frac{1}{n_1 + 1} = \eta\frac{n_1}{n_1 + 1} < \eta < 1 \tag{10}$$

$$\eta_d = \eta - \eta\frac{1}{n_1 + 1} = \eta\frac{n_1}{n_1 + 1} > 0 \tag{11}$$

Equations (10) and (11) denote the new learning rate $\eta_d \in (0, 1)$. Equation (10) denotes that η_d is getting smaller. □

Remark 2. With the iteration of (9), η_i is getting larger and can be used as a new learning rate whose value requires to be in the interval (0,1).

Proof. Given original learning rate $\eta \in (0, 1)$, the corresponding number of frames $n_2 \in N^+$ meets condition 2 in Table 1. According to Eq. (9), a new learning rate η_i is updated as $\eta + (1 - \eta)(1/(n_2 + 1))$. Therefore, we have two inequalities as follows:

$$\eta_i = \eta + (1 - \eta)\frac{1}{n_2 + 1} = \frac{\eta n_2 + 1}{n_2 + 1} < \frac{n_2 + 1}{n_2 + 1} = 1 \tag{12}$$

$$\eta_i = \eta + (1 - \eta)\frac{1}{n_2 + 1} = \frac{\eta n_2 + 1}{n_2 + 1} > \frac{\eta n_2 + \eta}{n_2 + 1} = \eta > 0 \tag{13}$$

Equations (12) and (13) denote the new learning rate $\eta_i \in (0, 1)$. Equation (13) denotes that η_d is getting larger. □

Algorithm 1 presents an overall procedure of our proposed approach.

4. Experimental Results

In this section, implementation details of experiments are first discussed. Secondly, we perform multiple trackers with different features setting based on DSST to validate the effectiveness of our integrated features. Thirdly, we investigate the most suitable threshold for learning rate update and validate the effectiveness of skewness. Finally, we evaluate propose algorithm on benchmark dataset containing 50 sequence with comparison to 12 state-of-the-art algorithms.

4.1. Implementation Details. We perform the experiment in MATLAB R2015b on Intel (R) Core (TM) i7-6700K 4.00 GHz CPU. The regularization parameter λ and scale number a are set the same as DSST. The original learning rate is set to η=0.025. The threshold for learning rate update is set to

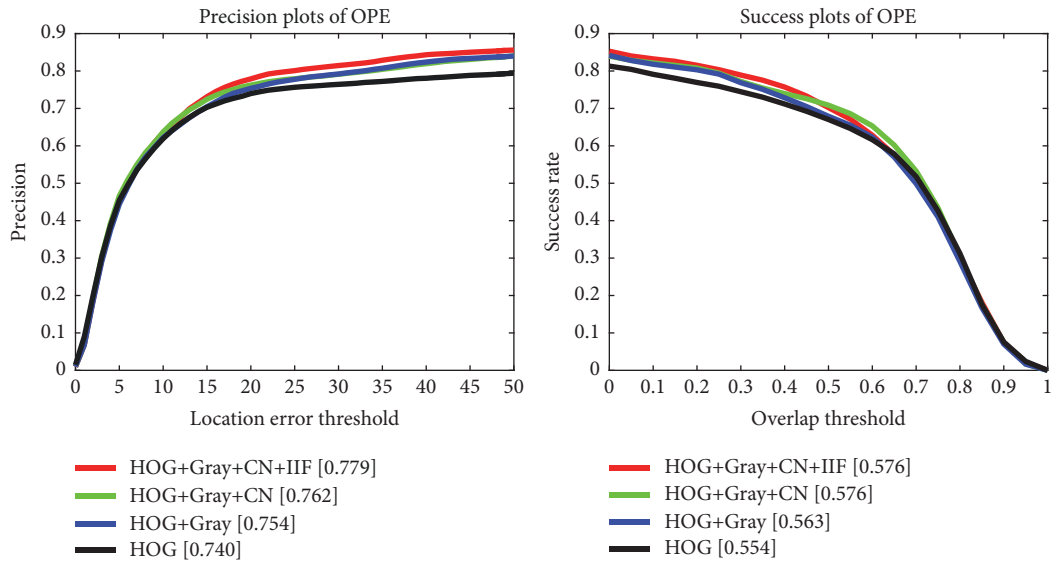


Figure 4: Success and precision plot using one-pass evaluation on the OTB. The performances of algorithms with variations in features.

θ=2.3. To prevent the boundary effect, the extracted features are usually multiplied by a Hann window. We use OTB-2013 dataset which contains 50 sequences and adopts one-pass evaluation (OPE) reporting by two aspects: precision and success plot. The precision plot shows the ratio of correct frames whose distance between predicted location of target and the ground truth not exceeds a certain threshold. We use center location error with a threshold of 20 pixels to rank tracking algorithms for precision plot. Success plot shows the ratio of correct frames whose overlap rate between prediction and the ground truth exceeds the given bounding box overlap threshold. For success plot, we rank tracking algorithm by employing the area under curve (AUC). In addition, we also utilize average speed to measure the efficiency of excellent algorithms.

4.2. The Multiple Feature Comparison. We implement several variations of our tracker to verify the validity of our approach. Figure 4 presents the tracking results with different features. As can be seen from Figure 4, our algorithm using the gray features, HOG, CN, and IIF achieves excellent performance in precision and performs as good as the variation tracker with three types of features (HOG, gray features, and CN) in success rate. The algorithm with three types of features (HOG, gray features, and CN) outperforms one with two types of features (HOG, gray features). The tracker with only HOG features has the worst performance among compared trackers. The results from our experiments indicate that the multiple feature integration is effective and robust.

4.3. The Detailed Analysis of Skewness

4.3.1. The Threshold Analysis. The threshold value has significant influence on the result of tracking and it is important to investigate the optimal threshold for learning rate update. The differences of two sequential frames on the *Basketball* and *Woman* sequences are shown in the Figure 5. By analyzing the skewness difference between two adjacent frames, we approximately estimate the range of the threshold and define threshold set as θ= {1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5}. We found that tracking performance gradually improved as threshold is gradually increased by 0.1. When the threshold is increased to 2.3 or 2.4, the experimental result is the best. But when the threshold increases again, the performance of the tracker begins to decrease. As is shown in Figure 6, we find that the tracker has the best robustness and accuracy when the threshold is 2.3 or 2.4, and we employ θ= 2.3 as a threshold value for our algorithm.

4.3.2. The Effectiveness Analysis of Skewness. To validate the effectiveness of our proposed skewness, we progressively incorporate our contribution. We add the integration features proposed in this paper into the baseline DSST and refer to it as multifeatures in Figure 7. Skewness is incorporated into the multifeatures, i.e., our tracker, which obtains the best results in precision and success rate. Figure 7 presents the performance results of our skewness. When the model update threshold is 2.3, the precision and success rate of our tracker have reached 79.9% and 58.4%, respectively. Compared with multifeatures, the precision and success rate increased by 2% and 0.8%, respectively. It illustrates that our model updating strategy which apply skewness model to decide the update of learning rate is effective. While compared with baseline DSST, our tracker increased by 5.9% in precision and 3% in success rate, respectively.

4.4. Comparisons with State-of-the-Art Trackers. We evaluate proposed tracker with 12 existing state-of-the-art trackers which include MIL [7], TGPR [36], Struck [10], CSK [16], KCF [18], DSST [19], ASLA [37], SCM [38], TLD [9], CT [8], DLT [27], and HDT [29]. Specifically, DLT and HDT are methods based on deep learning. The quantitative, attribute-based, efficiency, and qualitative evaluations are implemented in this section.

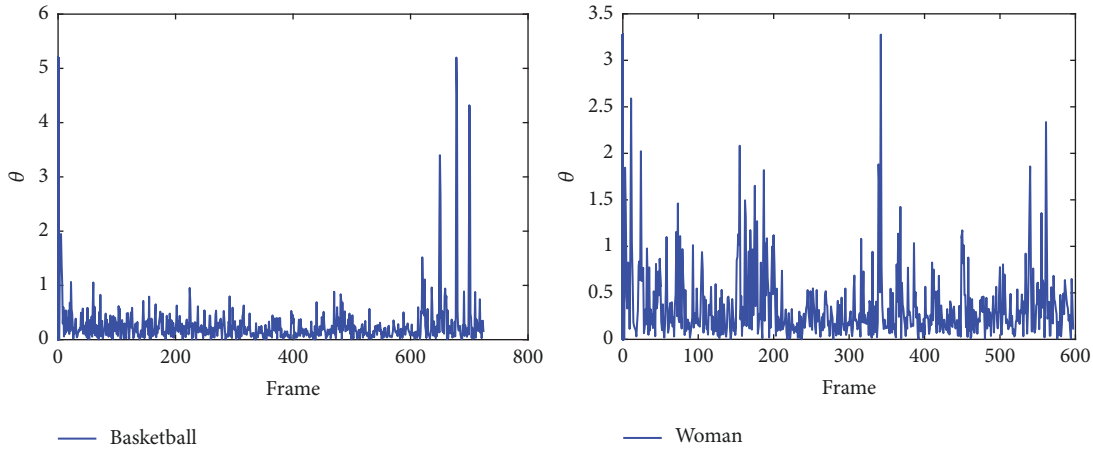


Figure 5: Analysis of differences between two adjacent frames on the *Basketball* and *Woman* sequences.

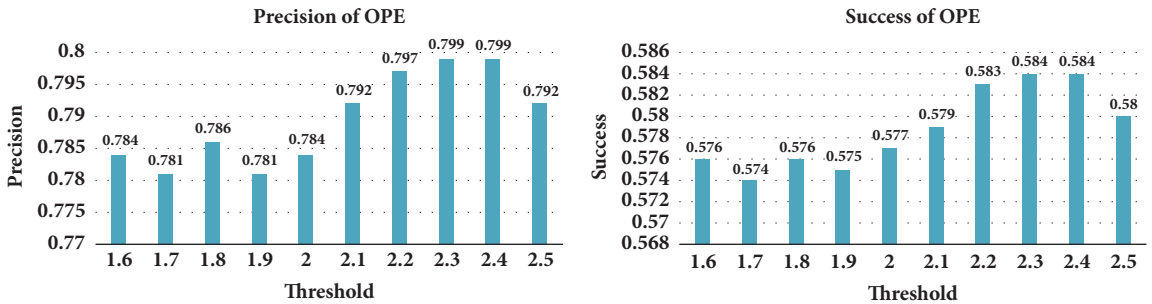


Figure 6: Comparisons of different thresholds in our algorithm.

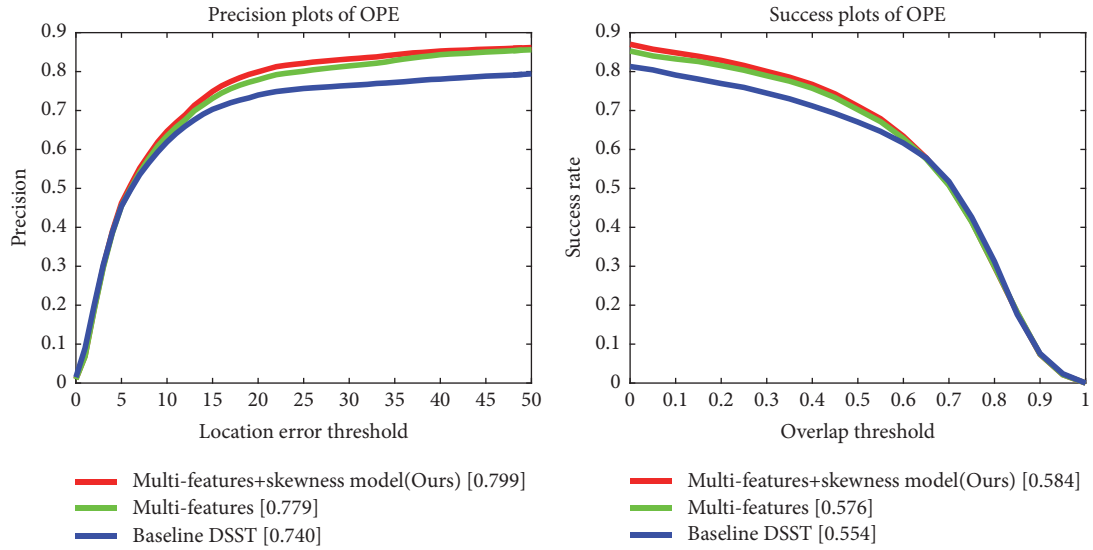


Figure 7: The performance results of skewness on OTB-2013.

4.4.1. Quantitative Evaluation. Figure 8 shows one-pass evaluation (OPE) results on 50 sequences. Our tracker performs well in both precision and success rate and just behind HDT. HDT incorporates deep features into correlation filter framework, which enhance the ability of target representation. Compared with it, the performance of our approach falls behind, but the speed of our tracker is far ahead of HDT. DLT is also based on deep learning, but its performance in precision and success plots is inferior to our tracker. The baseline framework DSST ranks the fifth in precision and occupies the third place in success rate. Overall, our tracker is better than most existing excellent trackers.

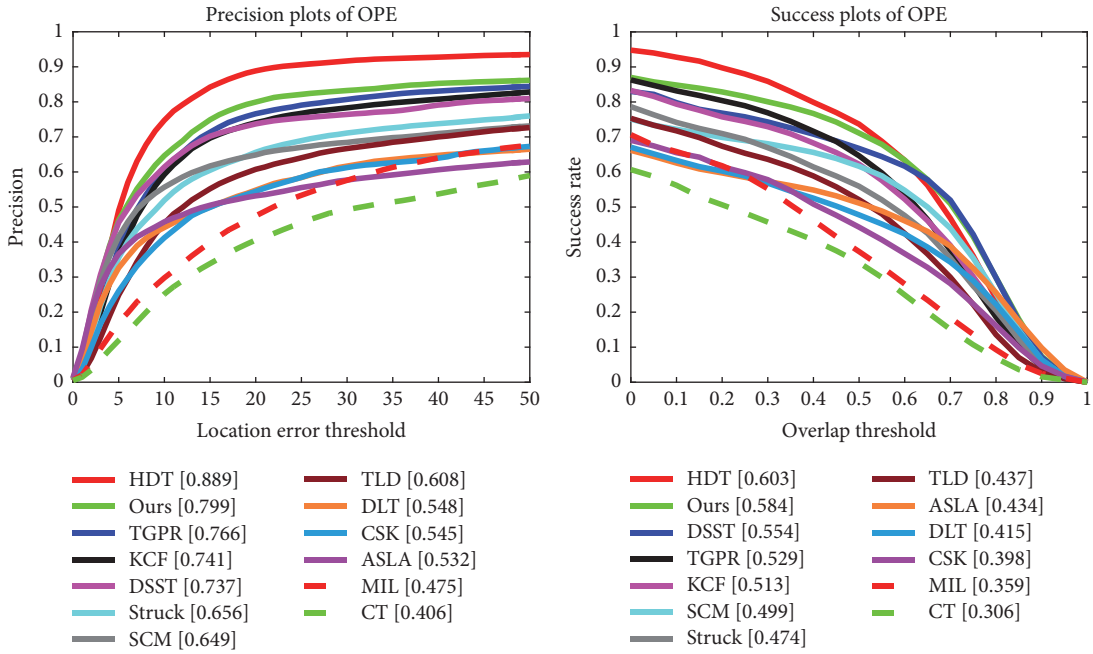


Figure 8: Evaluation results on OTB-2013.

4.4.2. Attribute-Based Evaluation. We compare our tracker with existing tracker based on 50 sequences annotated by 11 challenging attributes. Figures 9 and 10 illustrate attribute-based evaluations of all trackers in precision and success plots. Our algorithm favorably outperforms the most existing trackers in all challenging evaluations. The results show that our tracker is effective. In success plot, our tracker is superior to HDT in presence of illumination variations. It can be attributed to the fact that our fusion framework contains IIF features that are robust to severe illumination changes. Our algorithm outperforms other compared algorithms in occlusion, motion blur, and fast motion because our proposed model updating strategy plays an instructive role in these challenging sequences.

4.4.3. Efficiency Evaluation. We evaluate the operating efficiency of our proposed tracker with comparisons to 12 existing excellent trackers on 50 benchmark sequences. The results in Table 2 show that the average speeds of 13 trackers. Among these trackers, CSK achieves the best results with an average speed of 269.45 fps and KCF tracker acquires the second highest speed. Our proposed algorithm performs well with an average speed at 43.798 frames per second. DSST tracker achieves the average speed of 25.919 fps which are provided from [19], while the tracker obtains an average speed at 63.683 fps in our experimental platform. HDT obtains the slowest tracking speed because it is time-consuming to extract features from deep neural networks. Our integrated features and model updating strategy have a slight effect on tracking speed. However, our tracker is still faster than most of compared trackers.

4.4.4. Qualitative Evaluation. Figure 11 summarizes a qualitative comparison of proposed tracker with five existing excellent trackers (Struck [10], TGPR [36], KCF [18], DSST [19], and CSK [16]) on five challenging sequences. Comparison results of different algorithms are represented via solid rectangular frames with different colors. Five frames of each video sequence are selected to display the results and they contain common tracking problems. The *Sylvester* sequence contains illumination variations and out-of-plane rotation. The first row of Figure 10 shows partial tracking results of *Sylvester* sequence. The target suffers from rotation in the 935th frame. However, our algorithm is more adaptable to rotation than DSST, which mainly attribute to CN features. Although TGPR and Struck keep path with the target in the following sequence, their tracking effect is inferior to our

Table 2: Average speeds of 13 excellent trackers on 50 challenging image sequences.

Tracker	Speed(fps)
Ours	43.798
TLD [9]	21.742
MIL [7]	28.059
DSST [19]	25.919 (from [19]) 63.683 (our hardware)
Struck [10]	10.009
ASLA [37]	7.482
TGPR [36]	1.522
SCM [38]	0.374
CSK [16]	269.45
KCF [18]	245.87
CT [8]	28.79
HDT [29]	1.73
DLT [27]	15.00

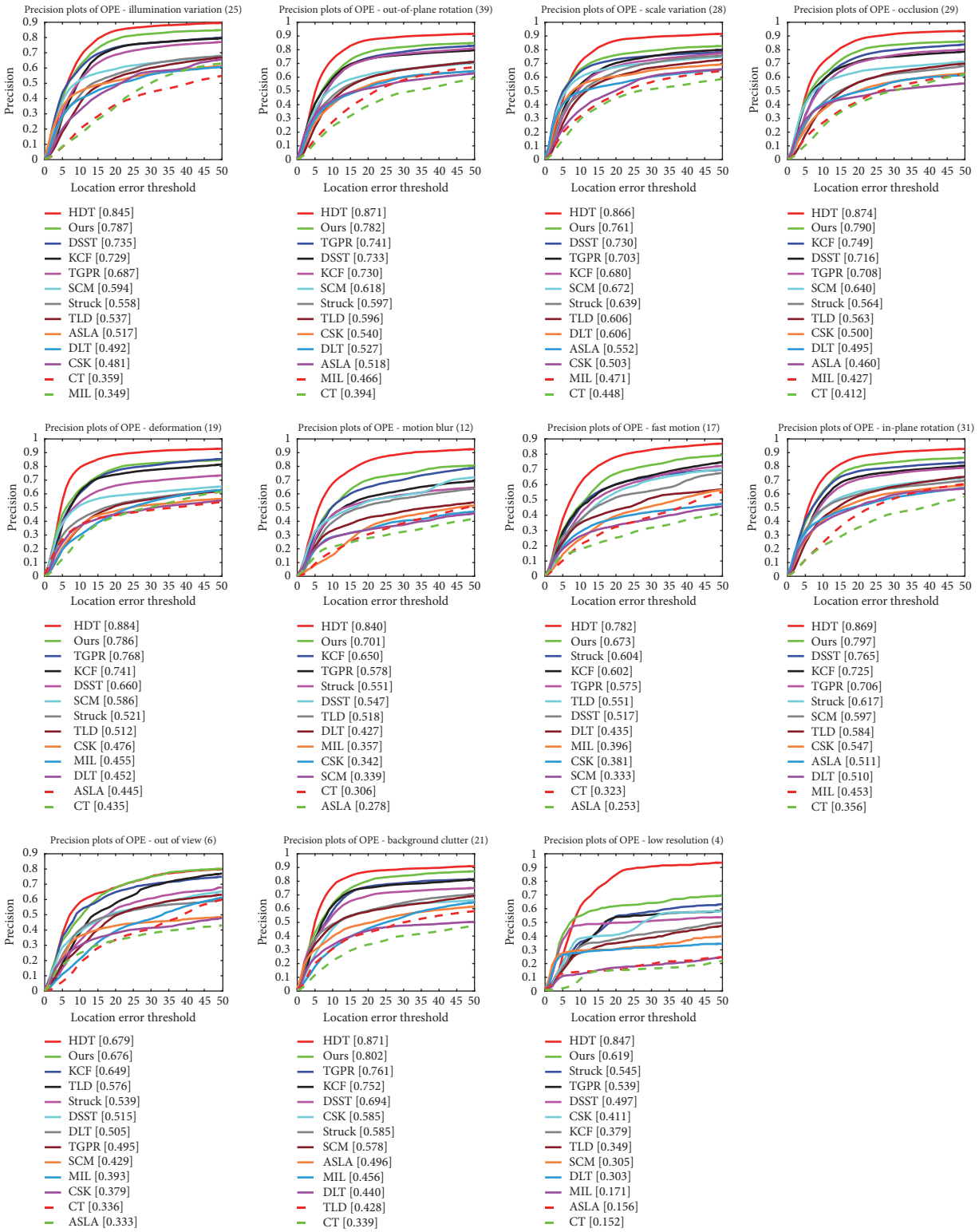


Figure 9: The precision plots show the attribute-based evaluation of our proposed tracker with 10 existing excellent trackers on 50 sequences. Our algorithm outperformed all trackers in all attributes.

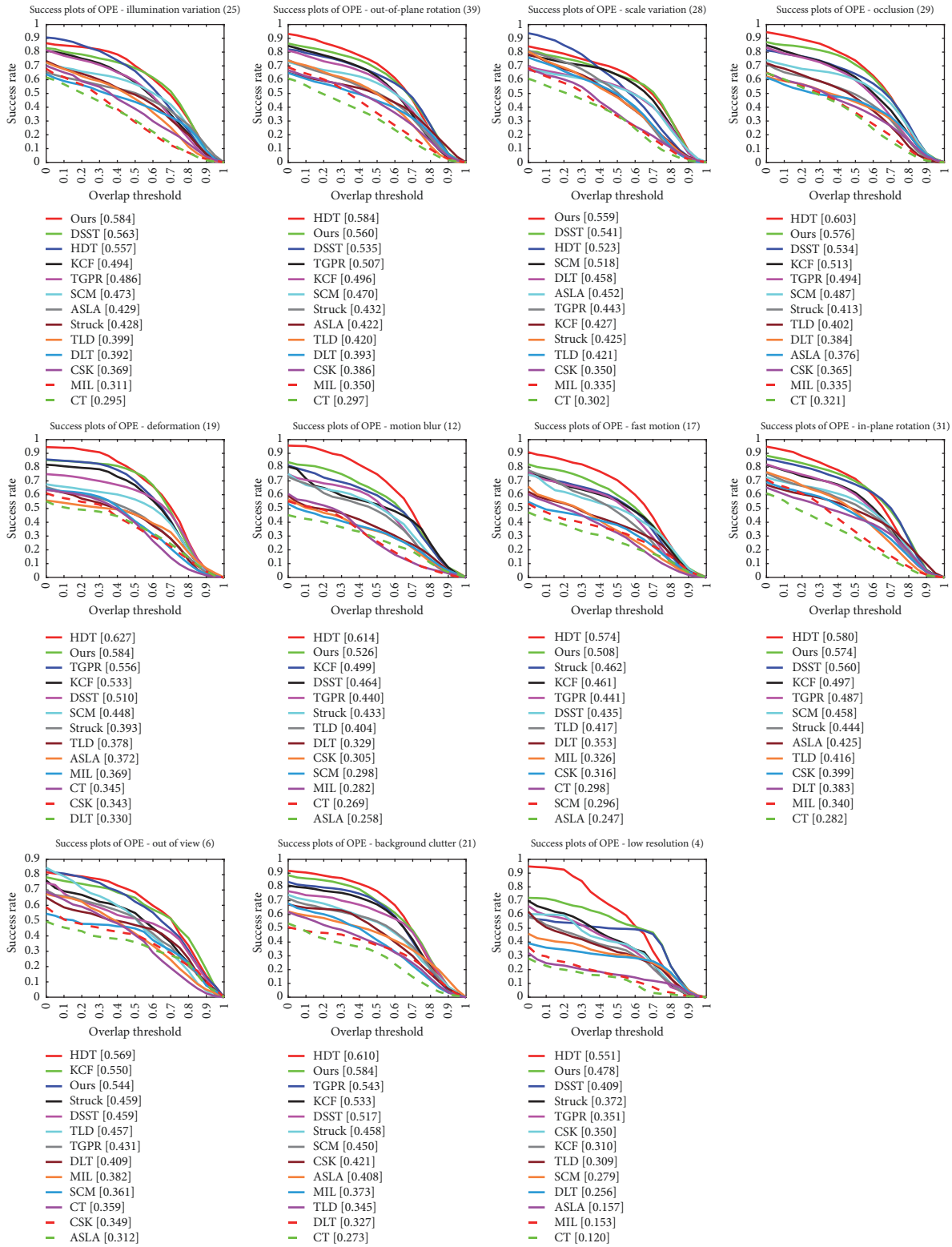


Figure 10: The success plots demonstrate our algorithm performance significantly better than existing excellent trackers in all tracking challenging.

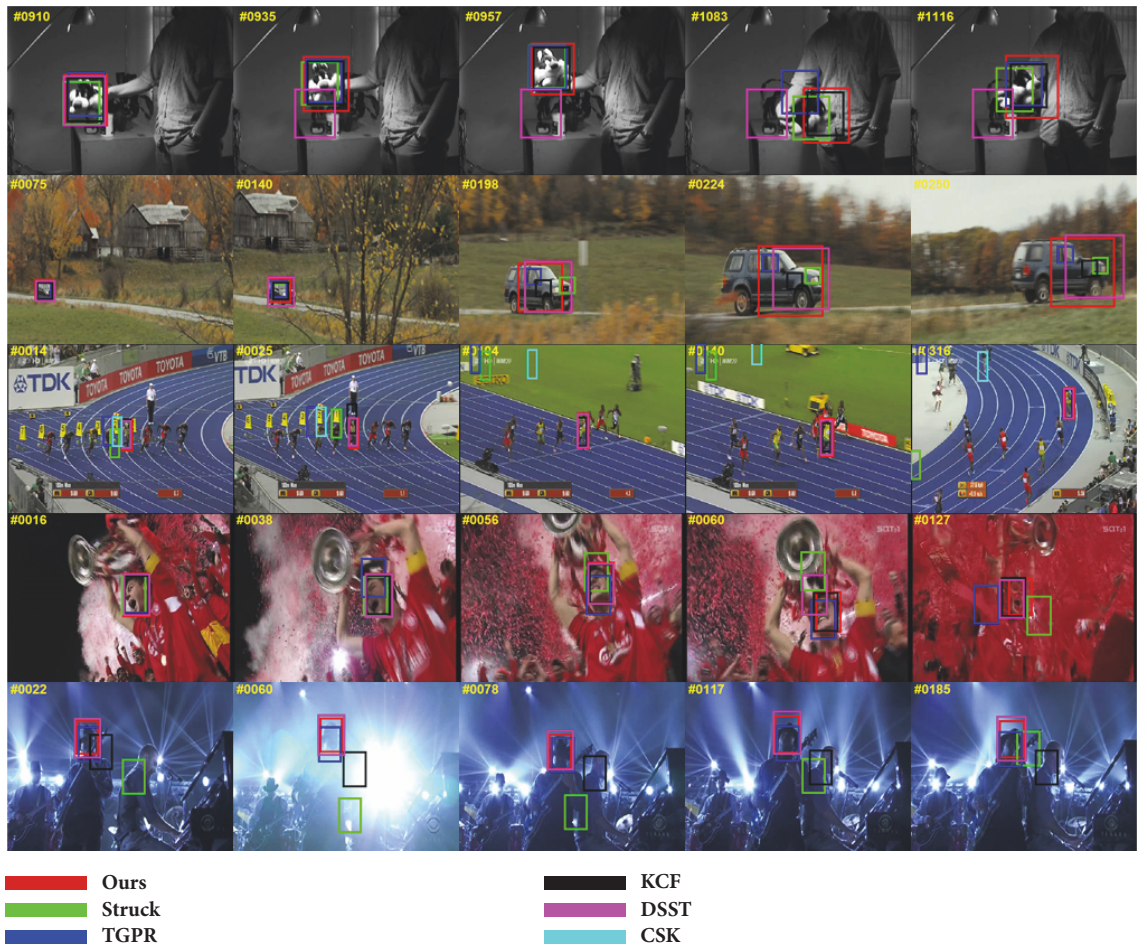


Figure 11: Tracking results comparison of our algorithm, Struck, TGPR, KCF, DSST, and CSK methods on five image sequences (from top to bottom, namely, *Sylvester*, *CarScale*, *Bolt*, *Soccer*, and *Shacking*, respectively).

tracker. The *CarScale* sequence shown in the second row presents scale variation. DSST and our tracker are better for tracking on this sequence, but they still can not completely mark the target. It is found that the reason for the phenomenon of tracking drift is that the target moves quickly and heavily occluded by the trees. The other four trackers suffer from heavily scale drift due to no adaptive scale estimation. The *Bolt* sequence comprises occlusion and deformation. Our tracker and DSST work well, while TGPR and Struck trackers lose the target due to the deformation and the following tracking is always in the state of tracking the wrong target. The *Soccer* sequence comprises occlusion, motion blur, fast motion, and illumination variation. Although our tracker and DSST tracker in this paper accurately track the target, the DSST tracker is still insufficient compared to our tracker. The reason is that our proposed model updating strategy has anti-interference ability in presence of motion blur. Target of TGPR tracker appears drifting in the 38th frame and TGPR fails to track the target in the 127th frame. It is unstable during the tracking process. *Shacking* sequence exhibits illumination variation and complex background. Our tracker and DSST can keep up with the target, but KCF and Struck trackers have poor performance in all frames of Shacking sequence. From the previous analysis, our tracker performance is superior to the above five state-the-art trackers in general.

5. Conclusion

In this paper, we put forward a simple and fast object tracker based on DSST. Our method extracts powerful features including gray features, HOG, CN, and IIF to learn correlation filters for estimating the target position and the scale is estimated by constructing feature pyramid. To prevent model drift, we further introduce skewness to measure the confidence degree of tracking results and update the learning rate by comparing the skewness value of two adjacent frames. Our tracker performs the excellent performance with the help of cooperation between the integrated features and dynamic learning rate in tracking. Contrastive experiments demonstrate that superiority of our proposed tracking algorithm over the 12 existing state-of-the-art algorithms on popular tracking benchmark dataset.

Data Availability

We are grateful to the Computer Vision Lab, Hanyang University, Seoul, Korea who provide the dataset publicly

(Visual Tracker Benchmark, http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html).

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The research work was funded by the National Natural Science Foundation of China grants nos. 61402053, 61772454, and 61811530332, the Scientific Research Fund of Hunan Provincial Education Department Grant no. 16A008, the Industry-University Cooperation and Collaborative Education Project of Department of Higher Education of Ministry of Education Grant no. 201702137008, the Postgraduate Scientific Research Innovation Fund of Hunan Province Grant no. CX2018B565, the Undergraduate Inquiry Learning and Innovative Experimental Fund of CSUST Grant no. 2018-6-119, and the Postgraduate Training Innovation Base Construction Project of Hunan Province Grant no. 2017-451-30.

References

[1] X. Lan, A. J. Ma, P. C. Yuen, and R. Chellappa, "Joint sparse representation and robust feature-level fusion for multi-cue visual tracking," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5826–5841, 2015.

[2] X. Lan, S. Zhang, P. C. Yuen, and R. Chellappa, "Learning common and feature-specific patterns: a novel multiple-sparse-representation-based tracker," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2022–2037, 2018.

[3] X. Lan, M. Ye, S. Zhang, H. Zhou, and P. C. Yuen, "Modality-correlation-aware sparse representation for RGB-infrared object tracking," *Pattern Recognition Letters*, 2018.

[4] B. Han, D. Comaniciu, Y. Zhu, and L. S. Davis, "Sequential kernel density approximation and its application to real-time visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1186–1197, 2008.

[5] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[6] 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.

[7] B. Babenko, M. 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, 2010.

[8] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proceedings of the 12th European Conference on Computer Vision (ECCV '12)*, pp. 864–877, 2012.

[9] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[10] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[11] S. Hong, T. You, and S. Kwak, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proceedings of the In Proceedings of the 32nd International Conference on Machine Learning, (ICML, 2015, Lile*, pp. 597–606, France, 2015.

[12] J. Ning, J. Yang, S. Jiang, L. Zhang, and M. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 4266–4274, Las Vegas, Nev, USA, June 2016.

[13] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Computer Vision—ECCV 2008: Proceedings of the 10th European Conference on Computer Vision, Marseille, France, October 12–18, 2008, Part I*, vol. 5302 of *Lecture Notes in Computer Science*, pp. 234–247, Springer, Berlin, Germany, 2008.

[14] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the British Machine Vision Conference (BMVC '06)*, pp. 47–56, September 2006.

[15] 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.

[16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision—ECCV 2012*, vol. 7575 of *Lecture Notes in Computer Science*, pp. 702–715, Springer, Berlin, Germany, 2012.

[17] 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.

[18] 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.

[19] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg, "Accurate Scale Estimation for Robust Visual Tracking," in *Proceedings of the British Machine Vision Conference*, pp. 65.1-65.11, Nottingham, UK, 2014.

[20] 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.

[21] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Computer Vision—ECCV 2014 Workshops*, vol. 8926 of *Lecture Notes in Computer Science*, pp. 254–265, Springer, 2015.

[22] X. Lan, Z. Shengping, and P. C. Yuen, "Robust joint discriminative feature learning for visual tracking," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016*, pp. 3403–3410, July 2016.

[23] X Lan, Ye Mang, S Zhang, and P. C. Yuen, "Robust Collaborative Discriminative Learning for RGB-Infrared Tracking," in *Proceedings of the In Proceedings of 32th AAAI Conference on Artificial Intelligence (AAAI*, pp. 7008–7015, New Orleans, Louisiana, US, 27, February 2018.

[24] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR '16), pp. 1401–1409, IEEE, Las Vegas, Nev, USA, June 2016.

[25] Y. Tu, Y. Lin, J. Wang, and J. Kim, "Semi-supervised Learning with Generative Adversarial Networks on Digital Signal Modulation Classification," *Computers Materials & Continua*, vol. 55, no. 2, pp. 243–254, 2018.

[26] D. Zeng, Y. Dai, F. Li, R. S. Sherratt, and J. Wang, "Adversarial learning for distant supervised relation extraction," *Computers Materials & Continua*, vol. 55, no. 1, pp. 121–136, 2018.

[27] N. Wang and D. Y. Yeung, "Learning a Deep Compact Image Representation for Visual Tracking," in *Proceedings of the In Proceedings of 27th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV*, p. 10, 2013.

[28] C. Ma, J. B. Huang, X. K. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '15)*, pp. 3074–3082, Santiago, Chile, December 2015.

[29] Y. Qi, S. Zhang, L. Qin et al., "Hedged Deep Tracking," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4303–4311, Las Vegas, NV, USA, June 2016.

[30] J. Zhang, X. Jin, J. Sun, J. Wang, and A. K. Sangaiah, "Spatial and semantic convolutional features for robust visual object tracking," *Multimedia Tools and Applications*, 2018.

[31] C. Ma, X. Yang, . Chongyang Zhang, and M. Yang, "Long-term correlation tracking," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5388–5396, Boston, MA, USA, June 2015.

[32] H. Sun, J. Li, J. Chang, B. Du, and Z. Su, "Efficient compressive sensing tracking via mixed classifier decision," *Science China Information Sciences*, vol. 59, no. 7, 2016.

[33] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization," in *Computer Vision – ECCV 2014*, vol. 8694 of *Lecture Notes in Computer Science*, pp. 188–203, Springer International Publishing, Cham, 2014.

[34] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.

[35] R Zabih and J Woodfill, *Computer Vision — ECCV '94*, vol. 801, Springer-Verlag, Berlin/Heidelberg, 1994.

[36] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8691 of *Lecture Notes in Computer Science*, pp. 188–203, 2014.

[37] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1822–1829, June 2012.

[38] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2356–2368, 2014.