

Research Article

Scale Adaptive Kernelized Correlation Filter Tracker with Feature Fusion

Tongxue Zhou,^{1,2,3} Ming Zhu,¹ Dongdong Zeng,^{1,2,3} and Hang Yang¹

¹Chinese Academy of Sciences, Changchun Institute of Optics, Fine Mechanics and Physics, Changchun 130033, China

²The University of the Chinese Academy of Sciences, Beijing 100049, China

³The Key Laboratory of Airborne Optical Imaging and Measurement, Chinese Academy of Sciences, Changchun 130033, China

Correspondence should be addressed to Tongxue Zhou; zhoutongxue1992@163.com

Received 13 May 2017; Revised 26 June 2017; Accepted 6 July 2017; Published 17 October 2017

Academic Editor: Mauro Gaggero

Copyright © 2017 Tongxue Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Visual tracking is one of the most important components in numerous applications of computer vision. Although correlation filter based trackers gained popularity due to their efficiency, there is a need to improve the overall tracking capability. In this paper, a tracking algorithm based on the kernelized correlation filter (KCF) is proposed. First, fused features including HOG, color-naming, and HSV are employed to boost the tracking performance. Second, to tackle the fixed template size, a scale adaptive scheme is proposed which strengthens the tracking precision. Third, an adaptive learning rate and an occlusion detection mechanism are presented to update the target appearance model in presence of occlusion problem. Extensive evaluation on the OTB-2013 dataset demonstrates that the proposed tracker outperforms the state-of-the-art trackers significantly. The results show that our tracker gets a 14.79% improvement in success rate and a 7.43% improvement in precision rate compared to the original KCF tracker, and our tracker is robust to illumination variations, scale variations, occlusion, and other complex scenes.

1. Introduction

Visual tracking is a challenging topic in the computer vision for its various applications in video surveillance, automatic driving, and medical fields. The aim of tracking is to predict target's position in video sequences, given its location in the first frame. Despite the fact that great progress has been made in recent years, designing a fast and efficient tracking is still difficult due to many reasons [1] like illumination change, occlusions, deformations, and scale variations.

In general, tracking algorithms can be divided into generative tracking and discriminative tracking. Generative tracking [2, 3] focuses on learning a target appearance model and it locates the target by searching the region that is most similar to the appearance model. It does not require a large dataset for training. However, search region is limited around the present position of target. Discriminative tracking [4, 5] addresses visual tracking as a classification problem. It learns from target and background and predicts region as target or background. It generally requires a large dataset to achieve good performances. Although great progress has been made

in the two categories of tracking algorithms, it remains a challenging task to generalize the target appearance model from a limited set of training samples. Recently, correlation filters (CF) [6] have made huge success in tracking due to their speed and localization accuracy. It is designed to produce high peaks for a given target in the frame and low or no peaks for nontarget. Henriques et al. [7] proposed the CSK tracker to explore the structure of the circulant patch to enhance the classifier by the augmentation of negative samples, which adopt the gray feature into the visual tracking. To further boost the performance of CSK tracker, Danelljan et al. [8] adopt the color-naming feature into the visual tracking task, which is a powerful feature for the color objects. Based on CSK, Henriques et al. [9] introduced the kernelized correlation filter (KCF) into the tracking application and adopted the HOG feature instead of raw pixel to improve both the accuracy and robustness of the tracker.

Although the above-mentioned trackers achieved the appealing results, three important aspects mainly limit their accuracy and robustness. First, the trackers apply one kind of feature for tracking. Single feature has its limitation dealing

Inputs:

- x**: training image patch;
- y**: regression target;
- z**: test image patch;
- p₀**: initial target position;

Outputs:

- p_i**: detected target position;

Training stage:

- (1) Compute the Gaussian kernel correlation of x with itself: k^{xx} using (3);

- (2) Compute coefficient $\hat{\alpha}$ using (2);

Position Detection:

- (3) Compute the response $f(z)$ using (4);
- (4) Find the target position p_i by maximizing $f(z)$;

Model Online Update:

- (5) Update the template using the fixed learning rate using (5).

ALGORITHM 1: The KCF algorithm.

with various changes in tracking. To handle more challenging problems, fused features [10–13] are used to tackle with different target variations. So we use fused feature to improve the robustness of the tracker. Second, these trackers use the fixed template size, which is unable to solve the scale variations. To solve the scale change issue, we sample the target with different scales and resize them into a fixed size. Third, learning rate is fixed in most of the existing correlation filters [14]. In this paper, we proposed an adaptive learning rate to track the target and give an occlusion detection mechanism.

Our contributions in this work can be summarized as follows. First, fused features are used, including HOG, color-naming, and HSV to further boost the tracking performance. Second, we extend the KCF tracker with the capability of handling scale changes by using a scale adaptive scheme, which can strengthen the tracking precision. Third, we present an adaptive learning rate and an occlusion detection mechanism to update the target appearance model in presence of occlusion problem. Finally, the competitive results of OTB-2013 demonstrate that our proposed approach achieves performance gains in accuracy and robustness, compared to state-of-the-art trackers.

The rest of this paper is organized as follows. Section 2 describes the KCF algorithm. Section 3 presents our proposed tracking algorithm. Section 4 describes the experiment results from OTB-2013 dataset. Finally, Section 5 concludes the whole paper.

2. The KCF Algorithm

The KCF tracker achieves the fastest and most satisfying performance among the recent top-performing trackers and its principle is simple. In the following, we briefly introduce the KCF tracker. The overall tracking procedure is summarized into Algorithm 1. More details can be found in [9].

The KCF tracker casts the tracking problem as a classification problem. The classifier $f(x) = \langle \phi(x_i), w \rangle$ is trained on

a $M \times N$ image patch x centered around the target. The patch is twice larger than the size of the target. The KCF tracker uses a circulant matrix to learn all the possible shifts of the target; the cyclic shift versions x_i are considered as training samples, where $i \in \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\}$. The matching score $y_i \in [0, 1]$ is generated by a Gaussian function, and the classifier is trained by minimizing the ridge regression error:

$$w = \arg \min_w \sum_i |\langle \phi(x_i), w \rangle - y_i|^2 + \lambda \|w\|^2, \quad (1)$$

where $\phi(x)$ is the mapping function to a Hilbert space and $\lambda \geq 0$ is the regularization parameter controlling the model simplicity.

Defining the kernel $\phi^T(x)\phi(x') = k(x, x')$, the solution w can be derived as $w = \sum_i \alpha_i k(x, x')$. $\hat{\alpha}$ can be obtained from

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda}, \quad (2)$$

where k^{xx} is a vector whose i th element is $k(x_i, x)$. “ \leftarrow ” denotes the Discrete Fourier Transform (DFT).

For an image data with c feature channels, a concatenation $x = [x^1; \dots; x^c]$ can be constructed, and the kernel correlation k^{xx} can be computed with element-wise products in the Fourier domain. Thus, we have

$$k^{xx'} = \exp \left(-\frac{1}{\sigma^2} \left(\|x\|^2 + \|x'\|^2 - 2F^{-1} \left(\sum_{c=1}^c \hat{x}'_c \odot \hat{x}_c^* \right) \right) \right), \quad (3)$$

where \odot denotes the operator of element-wise products, and c is the index of the feature channels.

During the tracking stage, a $M \times N$ candidate image patch z is cropped out in the new frame. The matching score of z can be evaluated via

$$f(z) = \mathcal{F}^{-1} \left(\hat{\alpha} \odot \hat{k}^{\bar{x}z} \right), \quad (4)$$

where $f(z)$ is the matching score for all the cyclic shift versions of z , the position of the target is estimated by finding the highest score, and $\bar{\alpha}$ and \bar{x} are learned coefficients and target appearance model.

During the update stage, the coefficient $\bar{\alpha}$ and the target appearance model \bar{x} should be updated via

$$\begin{aligned} \hat{\bar{\alpha}}_t &= \eta \hat{\bar{\alpha}}_t + (1 - \eta) \hat{\bar{\alpha}}_{t-1} \\ \hat{\bar{x}}_t &= \eta \hat{\bar{x}}_t + (1 - \eta) \hat{\bar{x}}_{t-1}, \end{aligned} \quad (5)$$

where η is a fixed learning rate; it is usually set as 0.02.

3. The Proposed Tracking Algorithm

In this section, we describe our tracking process based on the kernelized correlation filters in detail. The overall tracking is summarized into Algorithm 2.

Our tracker begins with the initial bounding box, which locates the position of the tracking target in the first frame.

Inputs:

x : training image patch;
 y : regression target;
 z : test image patch;
 p_0 : initial target position;
 s_0 : initial target scale;

Outputs:

p_i : detected target position;
 s_i : detected target scale;

Training stage:

(1) Compute the Gaussian kernel correlation of x with itself: k^{xx} using (3);
(2) Compute coefficient $\hat{\alpha}$ using (2);

Position Detection:

(3) Compute the response $f(z)$ using (4);
(4) Find the target position p_i by maximizing $f(z)$;

Scale Prediction:

(5) Sample the new patch z^{k_i} based on size $k_i s_i$ and resize it to s_i ;
(6) Find the target scale $k_i s_i$ by computing the response $\hat{f}(z^{k_i})$ with (6);

Model Online Update:

(7) Update the template using (5) according to occlusion detection strategy.

ALGORITHM 2: The proposed tracking algorithm.

Then it extracts the fused feature and trains the RLS classifier to get a position correlation filter. And it locates the target by finding the maximum response of the correlation filter. Next we train another RLS classifier to get a scale correlation filter with multi-scale image patches and fused features. And get the optimal scale of the target in the new frame by finding the maximum response of the correlation filter. In the whole process, we update the filter template according to whether the target is occluded or not.

In the following subsections, we will introduce our proposed strategies: feature fusion, scale adaptive scheme, and model online update scheme.

3.1. The Feature Fusion. In recent years visual tracking is considered as a classification problem, where the goal is to distinguish target from local background. It requires extracting the best features that can separate target from background.

Since the KCF tracker only used the HOG features, we employ fused features to boost the tracking performance. Because only the dot product and vector norm are needed in nuclear related functions, the various features of the target can be considered as a multidimensional vector $x = [x_1, x_2, x_3, \dots, x_C]$, according to the property of multi-channel KCF algorithm; see (3); HOG, CN, and HSV features are employed to get the fused feature.

Histogram of Gradient (HOG) is one of the most popular features in visual tracking. The HOG features are 32-dimensional, including 18 contrast sensitive orientation channels, 9 contrast insensitive orientation channels, 4 texture channels, and 1 all zeros channel. The HOG features are robust to illumination and deformation.

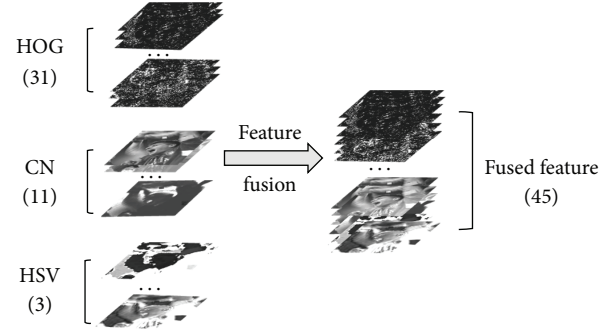


FIGURE 1: Process of the feature fusion.

Color-naming is a low dimensional adaptive extension of color attributes, which is the linguistic color label assigned by human to describe the color. RGB color is mapped to 11 basic color names, black, brown, gray, green, orange, pink, purple, red, blue, white, and yellow, which usually contains important information about the target.

HSV is also a color space; it includes hue, saturation, and intensity information. It is more in accord with human visual characteristics. HSV color space has better performance than RGB color space in visual tracking.

The three features are complementary to each other. We considered HOG features for gradient details, CN color space for color information, and HSV space for more detailed information. The HOG features are 31-dimensional (except the all zeros channel), the CN features are 11-dimensional, and the HSV features are 3-dimensional. Three features are employed to get the 45-dimensional integrated features, so the channel of the features is 45 in (3). Figure 1 shows the process of the feature fusion.

The response maps of single feature and fused feature are shown in Figure 2. As we can see the response map of the single feature is indistinguishable with much noise around the center but the response map of fused feature is more discriminative, which enhances the response area where all three maps have high confidence.

3.2. The Scale Adaptive Scheme. In visual tracking, scale change is one of the most common challenging aspects, which influences the tracking accuracy. The KCF tracker is unable to deal with the scale changes. In this section, we proposed an effective scale adaptive scheme.

For most tracking approaches, the template size of the target is fixed. In order to handle the scale variations, we proposed enlarging the scale space from the countable integer space into uncountable float space. Suppose that the template size is s_t in the original image space, and we define a scaling pool $S = \{k_1 s_t, k_2 s_t, \dots, k_k s_t\}$. For the current frame, we first sample k patches of the size in S . Note that the operation in the kernel correlation function needs the data with the same dimensions; we resize these patches by using the bilinear interpolation to the size of the initial target s_t before extracting the features. The process is shown in Figure 3.

Then we train another RLS classifier R_t to get a scale correlation filter on multi-scale image patches to estimate the

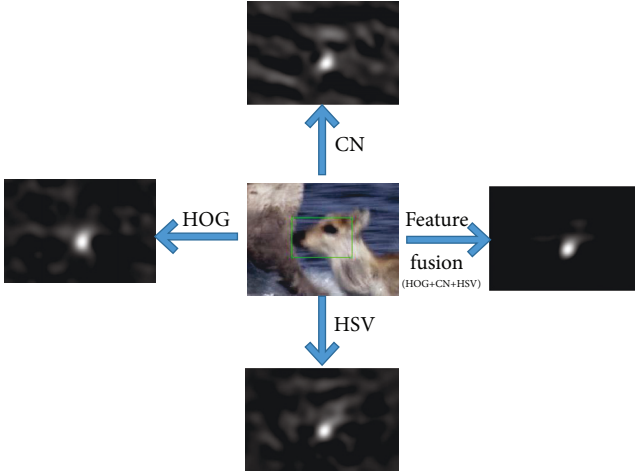


FIGURE 2: Response map of single feature and fused feature.

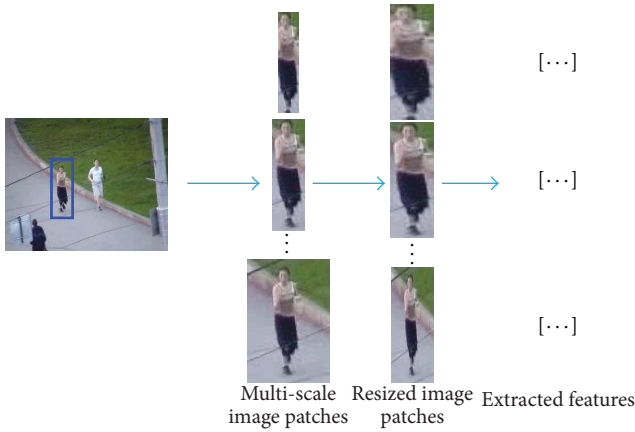


FIGURE 3: The scale adaptive scheme.

target scale. The training process of classifier R_t is similar to classifier R_c in KCF tracker and the target scale in a new frame is computed by finding the maximum response of the correlation filter.

$$\arg \max F^{-1} \hat{f}(z^{k_i}), \quad (6)$$

where z^{k_i} is the sample patch; its size is $k_i s_t$.

We should note that the target displacement is implied in the response map, so the final target position should be tuned by the scale factor k_i .

3.3. Model Online Update Scheme. During the visual tracking the appearance of the target often changes, so it is necessary to update the target appearance model to adapt to these changes [15, 16]. Because of appearance variations, training samples of the tracked targets collected by the online tracker are required for updating the tracking model. However, this often leads to tracking drift problem because of potentially corrupted samples, contaminated/outlier samples resulting from large variations (e.g., occlusion), which has been shown in [17]. And in most of the existing correlation filter based trackers,

the learning rates are usually fixed. Fixed learning rate limits the tracker's ability to adapt to quick changes of appearance, especially in the presence of occlusion. So we proposed a novel model online update scheme to track the target.

When the target is occluded, the appearance model should not be changed; the learning rate is set to 0. If the target is not occluded, the target appearance model will update with the normal learning rate. The learning rate is set as follows:

$$\eta = \begin{cases} 0.02, & \text{without occlusion} \\ 0, & \text{occlusion.} \end{cases} \quad (7)$$

The strategy to judge whether the target is occluded is as follows:

(1) According to (4), we obtain the target's position $\text{pos}_{\max}(z)$ by maximizing $f(z)$;

(2) We get the number of the positions, Num, whose responses satisfy the relationship $f(z) > \beta_1 \cdot f_{\max}(z)$;

(3) Then we judge whether the target is occluded by making a comparison between Num and $\beta_2 \times \text{width} \times \text{height}$, where width and height are the size of the sample.

If Num is larger than $\beta_2 \times \text{width} \times \text{height}$, then the target is occluded; the learning rate is set to 0. If Num is smaller, the target is not occluded; the target appearance model will update with the normal learning rate $\eta = 0.02$.

Suppose that the target in the presence of occlusion can be considered to mix with a Gaussian noise. Because the occlusion area also does circular shift operation, when it performs convolution with the classifier model parameter α (see (4)), the response values around $f_{\max}(z)$ increase and the distribution of $f(z)$ tends to be smooth.

As shown in Figure 4, (b) and (d) are the response maps of the tracking target in (a) and (c), respectively. The brighter pixel corresponds to the greater probability value; we can see that when the target is not occluded the response map of $f(z)$ is discriminative and when the target is occluded the response map is vague. So the distribution of $f(z)$ is consistent with our hypothesis and the occlusion detection strategy is reasonable.

4. Experiment

In order to evaluate the overall performance of the proposed tracker, first, we evaluate the proposed tracker in the way of OTB-2013 dataset [18]. Second, we compare our tracker with the state-of-the-art trackers. Finally, we provide the qualitative analysis of our approach with existing tracking methods.

4.1. Experiment Settings. We implemented the proposed tracker by using MATLAB 2015A on Intel i5-5200U CPU, @2.20 GHz processor with 8 GB RAM. The parameters settings are as follows: σ used in Gaussian function is set to 0.5; the cell size of HOG is 4×4 . The scale pool $S = \{0.98, 0.99, 1.0, 1.01, 1.02\}$. In the occlusion detection strategy, the parameter $\beta_1 = 0.5$, $\beta_2 = 0.08$, and the learning rate $\eta = 0.02$.

4.2. The OTB-2013 Dataset. The OTB-2013 dataset contains 50 fully annotated sequences. These sequences include many

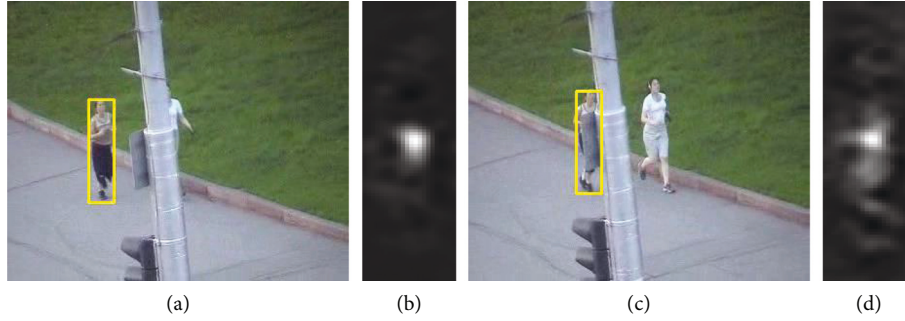


FIGURE 4: The original frames and response map of $f(z)$, respectively, (a) target without occlusion; (b) response map of $f(z)$ without occlusion; (c) target with occlusion; (d) response map of $f(z)$ in the presence of occlusion.

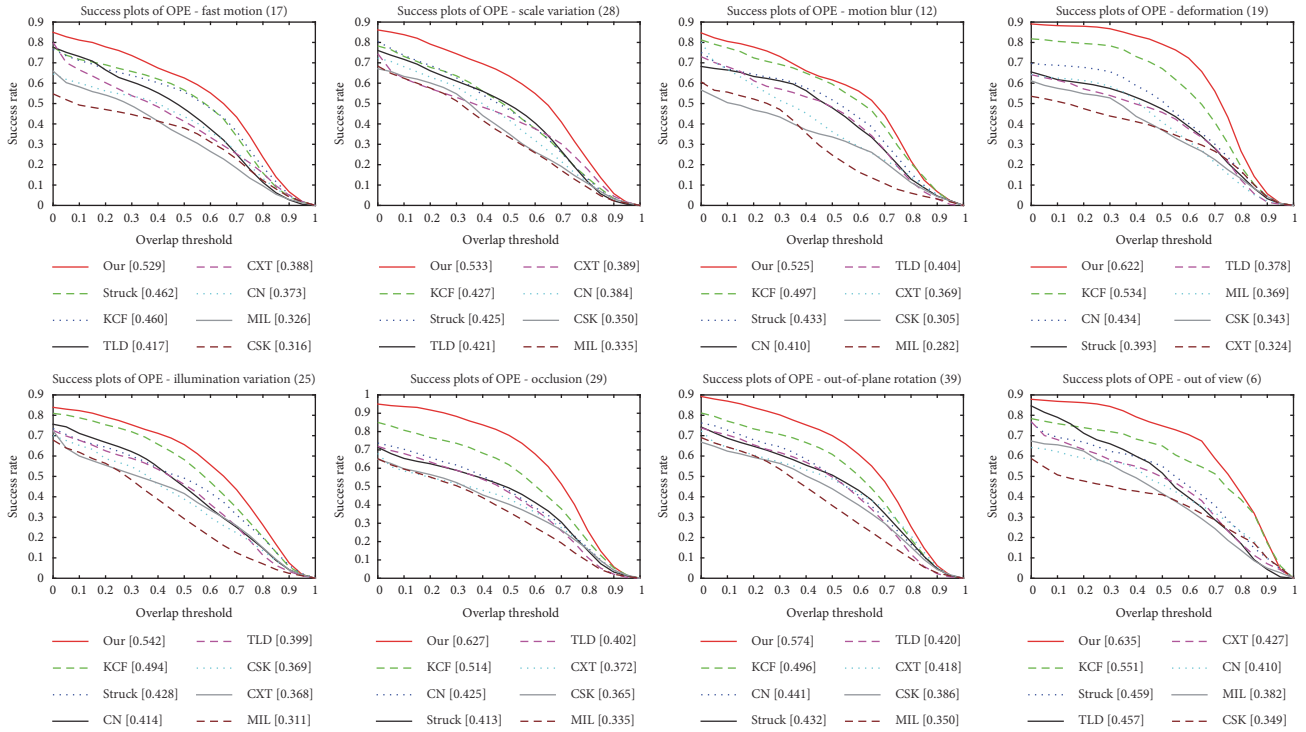


FIGURE 5: Success plots of videos with 8 different attributes for the top 8 trackers (the number in the title indicates the number of sequences).

popular sequences used in the online tracking literature over the past several years; these sequences are annotated with the 11 attributes including illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, and low resolution.

4.3. Performance Evaluation. To analyze the performances of different algorithms, the three evaluation metrics are used: Center location error (CLE), distance precision (DP), and overlap precision (OP). The first metric, CLE, is computed as the average Euclidean distance between the ground-truth and the estimated center location of the target. The second metric, DP, is computed as the percentage of frames in the sequence where the center location error is smaller than a certain threshold. The DP values are at a threshold of 20 pixels. The third metric, OP, is defined as the percentage of

frames where the bounding box overlap surpasses a threshold $t \in [0, 1]$. The OP values are at a threshold of 0.5, which correspond to the PASCAL evaluation criteria. What is more, the precision plots based on the location error metric and the success plots based on the overlap metric are adopted.

Three kinds of evaluation strategies are performed: one pass evaluation (OPE), temporal robustness evaluation (TRE), and spatial robustness evaluation (SRE). TRE randomizes the starting frame and runs a tracker through the rest of the sequences, and SRE randomizes the initial bounding boxes by shifting and scaling.

4.4. Experiment 1: Attribute Based Comparison. We evaluate our tracker for eight main challenging attributes. These attributes are used for analyzing the performance of trackers in different aspects. The success rate of 8 tackers on each attribute is shown in Figure 5. In the experiments, we can

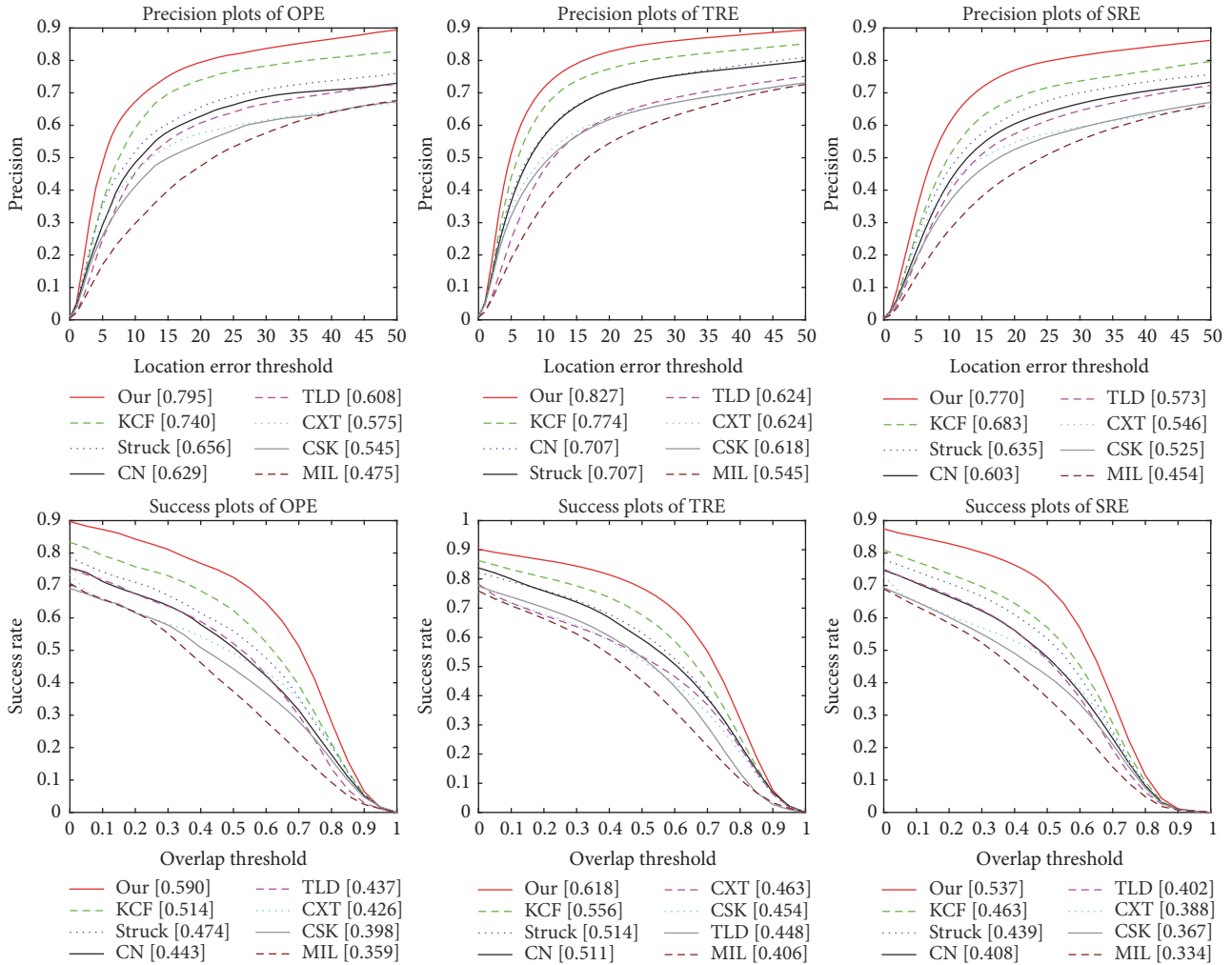


FIGURE 6: Precision plots and success plots of OPE, TRE, and SRE for the top 8 trackers.

observe that our tracker achieves the best performance among the 8 trackers; our tracker performs well with overall success rate in fast motion (52.9%), scale variation (53.3%), motion blur (52.5%), deformation (62.2%), illumination variation (54.2%), occlusion (62.7%), out-of-plane rotation (57.4%), and out-of-view (63.5%) while the KCF tracker achieves success rate of 46%, 42.7%, 49.7%, 53.4%, 49.4%, 51.4%, 49.6%, and 55.1%, respectively. In summary, our tracker achieves the best results in almost all the attributes.

4.5. Experiment 2: Comparison with the State-of-the-Art Trackers. Figure 6 illustrates a comparison with other 7 state-of-the-art methods on the OTB-2013 dataset. The 7 trackers are KCF [9], Struck [19], CN [8], TLD [20], CXT [21], CSK [6], and MIL [22]. We use precision plots and success plots on the term of OPE, TRE and SRE over all 50 sequences. From the success plots of OPE, we can see that our tracker achieves the best performance with average overlap threshold 0.590 which gets a 14.79% improvement upon KCF (0.514). From the precision plots of OPE, our tracker (0.795) gets a 7.43% improvement upon KCF (0.740). Since our model is based on KCF, the results show the robustness of our

tracker. To give sufficient comparison results, we also show the overall performance on TRE and SRE. For the results of TRE, our tracker gets an 11.15% of success plot and 6.85% of precision plot improvement, respectively, upon KCF. The results on TRE show the robustness of our tracker on initialization. For the results of SRE, comparing to KCF, our tracker gets a 15.98% of success plot and 12.74% of precision plot improvement, respectively, upon KCF.

4.6. Experiment 3: Qualitative Analysis. To evaluate the performance of our tracker, we run other three state-of-the-art trackers based on correlation filters (KCF, CN, and CSK) on 9 challenging sequences in Figures 7, 8, and 9. These sequences include three attributes: scale variation, illumination variation, and occlusion, which can validate the effectiveness of our tracker.

To validate the performance of our proposed scale prediction strategy, we choose three scale variation data sequences. These sequences from top to down are Boy, Dudek1, and Carscale, respectively. We compare our tracker with CSK, CN, and KCF. These three trackers use the fixed template size, while our tracker uses adaptive scale to track. The result is



FIGURE 7: The results on three scale variation sequences.

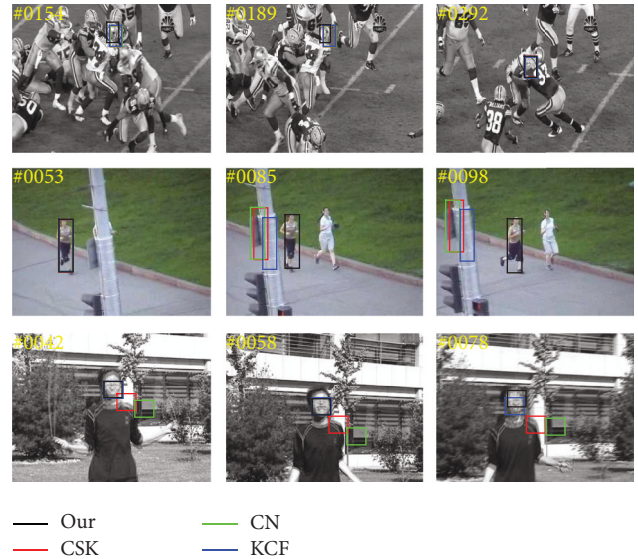


FIGURE 9: The results on three occlusion sequences.

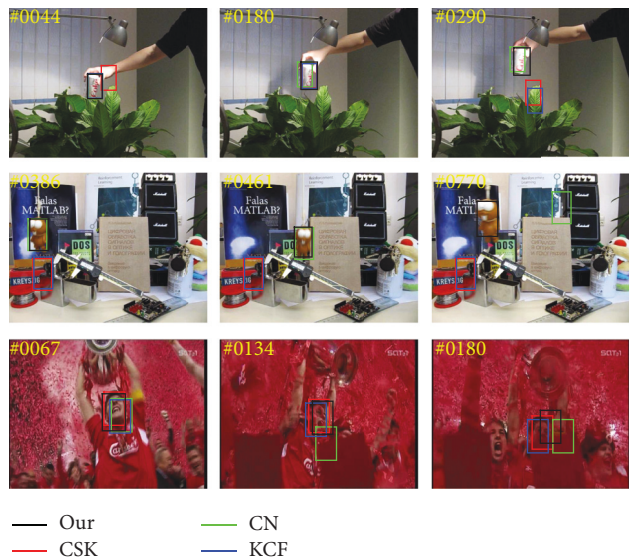


FIGURE 8: The results on three illumination variation sequences.

given in Figure 7. As we can see, our tracker performs well in these three sequences (Boy, Dudek1, and Carscale). The CN tracker drifts in the presence of abrupt motion (Boy). The KCF and CSK trackers perform well in Boy sequences, but cannot estimate scale changes (Dudek1 and Carscale). Our tracker is able to estimate scale changes. So the result shows the effectiveness of our adaptive scale scheme.

To validate the effectiveness of the fusion of features, we compare our tracker on three illumination variation data sequences. These sequences from top to down are Coke, Lemming, and Soccer, respectively. We compare our tracker with CSK, CN, and KCF; these three trackers only use single feature. CSK only employs the raw pixels, CN uses color-naming as feature, and KCF adopts HOG feature. Our tracker uses the fused features including HOG, color-naming, and

HSV. The result is shown in Figure 8. As we can see, our tracker performs well in handling illumination variation (Coke, Soccer, and Lemming) due to the representation of integrated features. However, the KCF and CSK trackers drift when target objects undergo illumination (Coke and Soccer). And the KCF, CN, and CSK trackers do not redetect targets in the case of tracking failure (Lemming). In summary, the fused features are effective and achieve promising tracking results.

To validate the effectiveness of our proposed adaptive learning rate method, we compare our tracker with CSK, CN, and KCF. For fair comparison, a fixed learning rate 0.02 is used in these three trackers, while our tracker uses an adaptive learning rate to track. We choose three sequences with the attribute of occlusion. These sequences from top to down are Football, Jogging-1, and Jumping, respectively. The result is shown in Figure 9. Our tracker performs well in all these sequences. Other trackers fail during heavy occlusion, because they use a fixed learning rate and cannot update the learning rate in time. And they cannot redetect the object in the case of tracking failure (Jogging-1). This suggests occlusion detection strategy and model online update scheme in our tracker play an important role in visual tracking.

Overall, our tracker performs well on these challenging sequences, which can be attributed to three reasons. First, our tracker is learned from fused features rather than single feature, so it is effective in tracking the targets. Second, due to the proposed effective scale adaptive strategy, our tracker is able to estimate target scale. Third, the occlusion detection strategy based on model online update scheme makes our tracker perform well when target objects undergo occlusion.

5. Conclusion

In this paper, we proposed a novel tracking algorithm based on the correlation filter. The fused features including HOG, color-naming, and HSV are used to boost the tracking performance. To deal with the scale changes, we proposed

an effective scale adaptive scheme; it shows its effectiveness on sequence with scale change. Besides, in order to adapt to the change of target, we employed a model online update scheme to update the target appearance model. The experiment results on 50 sequences demonstrate that our tracker outperforms the state-of-the-art trackers in terms of accuracy and robustness.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 61401425).

References

- [1] A. Smeulder, D. Chu, R. Cucchiara, S. Calderara, A. Deghan, and M. Shah, "Visual tracking: an experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.
- [2] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," in *Aerospace/Defense Sensing, Simulation, and Controls*, pp. 95–102, 2001.
- [3] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 142–149, June 2000.
- [4] H. T. Nguyen and A. W. M. Smeulders, "Robust tracking using foreground-background texture discrimination," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 277–293, 2006.
- [5] B. Babenko, S. Belongie, and M. Yang, "Visual tracking with online multiple instance learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 983–990, Miami, Fla, USA, June 2009.
- [6] A. Mahalanobis, B. V. K. V. Kuma, and D. Casasent, "Minimum average correlation energy filters," *Applied Optics*, vol. 26, no. 17, pp. 3633–3640, 1987.
- [7] F. 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 (ECCV)*, 2012.
- [8] 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, June 2014.
- [9] 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.
- [10] Y. Jiang and J. Ma, "Combination features and models for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 240–248, Boston, MA, USA, June 2015.
- [11] 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.
- [12] 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.
- [13] R. Liu, X. Lan, P. C. Yuen, and G. C. Feng, "Robust visual tracking using dynamic feature weighting based on multiple dictionary learning," in *Proceedings of the 24th European Signal Processing Conference, EUSIPCO 2016*, pp. 2166–2170, September 2016.
- [14] 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.
- [15] Z.-Q. Hou, A.-Q. Huang, W.-S. Yu, and X. Liu, "Visual object tracking method based on local patch model and model update," *Journal of Electronics and Information Technology*, vol. 37, no. 6, pp. 1357–1364, 2015.
- [16] G. An, H. Li, and M. Li, "Novel multi-window tracking based on model update," *IEEE International Symposium on Computational Intelligence & Design*, pp. 3–6, 2015.
- [17] X. Lan, P. C. Yuen, and R. Chellappa, "Robust MIL-based feature template learning for object tracking," in *Proceedings of the Conference on Artificial Intelligence*, pp. 4118–4125, 2017.
- [18] 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, OR, USA, June 2013.
- [19] 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.
- [20] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 49–56, June 2010.
- [21] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: exploring supporters and distracters in unconstrained environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1177–1184, June 2011.
- [22] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 983–990, Miami, FL, USA, June 2009.




Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

