Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 8659890, 9 pages
https://doi.org/10.1155/2020/8659890

Research Article

A Narrow Deep Learning Assisted Visual Tracking with Joint Features

Xiaoyan Qian and Daihao Zhang

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, 210016 Nanjing, China

Correspondence should be addressed to Xiaoyan Qian; qianxiaoyan@nuaa.edu.cn

Received 25 December 2018; Accepted 21 February 2019; Published 9 June 2020

Academic Editor: Wanquan Liu

A robust tracking method is proposed for complex visual sequences. Different from time-consuming offline training in current deep tracking, we design a simple two-layer online learning network which fuses local convolution features and global handcrafted features together to give the robust representation for visual tracking. The target state estimation is modeled by an adaptive Gaussian mixture. The motion information is used to direct the distribution of the candidate samples effectively. And meanwhile, an adaptive scale selection is addressed to avoid bringing extra background information. A corresponding object template model updating procedure is developed to account for possible occlusion and minor change. Our tracking method has a light structure and performs favorably against several state-of-the-art methods in tracking challenging scenarios on the recent tracking benchmark data set.

1. Introduction

Visual tracking is one important topic in computer vision with a wide range of applications, such as video surveillance, automobile navigation, human-computer interface, and driverless vehicle [1]. Although substantial progress has been proposed in recent years, it remains a challenging task due to many factors such as illumination changes, quick movement, and background disturbance [2].

To address these challenges for robust tracking, current visual tracking algorithms focus on exploiting robust handcrafted target representations, such as Haar-like features, color histogram, HOG descriptors, etc. Since each type of handcrafted feature is commonly able to address a few specific classical changes, they are not tailored for all generic objects and we require some sophisticated learning techniques to improve their representative capabilities. These learning methods build models to distinguish the target from the background. They typically learn classifiers based on multiple instance learning, online boosting, structured output SVMs, etc. Recently, correlation filter based tracking algorithms have achieved remarkable results due to the computational efficiency in the Fourier domain. The filter can locate the tracking target effectively, but all of them have the limitation of being excessively dependent on the maximum response value. When the response map becomes unreliable under some challenging circumstances, shift may occur even the object becomes lost. Different from handcrafted features, deep learning adopts hierarchical architecture to simulate human brain mechanism, which can generate outstanding representation for high nonstructured visual data. Convolutional neural network (CNN) for object recognition and detection has inspired tracking algorithms to employ the discriminative features learned by CNNs [3, 4]. To produce stable shared weights, CNN network needs a large number of training samples, while this is often not available in visual tracking as there exists only a few number of reliable positive instances extracted from the initial frame. In addition, because the CNNs are trained to recognize object classes, the deeper the network structure is, the faster the space information will lose. Thus, naively applying CNN models into tracking is not suitable. One way to address these problems is to fine-tune a pretrained CNN model. The other way is to design a narrow learning network.

Motivated by the challenges in object tracking in complex scenarios and inspired by the fusion method [5], we propose

a novel tracking scheme similar to correlation convolution, which takes advantage of convolutional and handcrafted features and meanwhile makes use of the adaptive Gaussian mixture filter (GMF) to generate samples effectively. The main contributions of this paper are summarized below:

(1) We propose an efficient feature extraction scheme, which combines local convolutional features and global handcrafted features to produce robust appearance expression. A simple narrow network is designed to generate high-level local features without pretraining. In this way, spatial information can be well preserved and it brings more accurate tracking.

(2) Our method takes advantage of strong correlation between adjacent frames to produce groups of convolution filters, which helps to discriminate the target from the surrounding background with the maximum likeness.

(3) Adaptive sampling scheme directs to reshape the candidates' distribution based on the 1-order motion information. This detection mechanism allows us to get the correct location of target object.

(4) In addition, we design an adaptive scale estimation method and an efficient model updating scheme. Spatiotemporal property decides scale changing and the updating degree.

The rest of this paper is structured as follows. We first review related work in Section 2. Next, the joint features are presented via a simple two-layer network. And the adaptive particle filter tracking model and the model updating scheme are described in Section 3. Section 4 demonstrates the objective and subjective experimental results in current benchmark datasets.

2. Related Work

Visual object tracking has been studied extensively and a comprehensive tracking can be found in [6–9]. In this section, we just review the works related to our method for simplicity, which include the particle filter based trackers and the deep learning based trackers.

PF algorithms have been studied in visual object tracking for many years and their variations are still widely used nowadays as it is neither limited to linear systems nor requires the noise to be Gaussian [10–12]. The traditional PF algorithm implements a recursive Bayesian framework by using the nonparametric Monte Carlo sampling method, which can effectively track target objects in most scenes. But challenging problems still exist. PF needs to design complex appearance models to deal with different visual sequences. And during updating the posterior distributions, it uses sequential importance sampling scheme to address the sample degeneration phenomenon when only a few particles representing the distribution have significant weights. Resampling may give limited results and is computationally expensive. Different from current PF, the appearance model can be automatically learned by convolutional network, which can directly be used in many challenging sequences. And we introduce motion information into Gaussian equation and the posterior distributions are directly decided by the state of the target objects without requiring particle resampling. Compared with current PF, it reduces computational complexity and brings adaptive particle distributions.

Deep neural networks are a powerful tool for learning image representations in computer vision applications. Inspired by the success of convolutional neural network in image classification and object recognition [13–15], researchers in tracking community have started to focus on the deep trackers that exploit the strength of CNN. These deep trackers come from two aspects: One trend is discriminative convolution trackers (DCT). It is the combination of excellent correlation filter tracking framework and CNN features. These tracking methods replace handcrafted features such as HOG with deep features and use correlation filter to find the maximum impulse [16–18]. The other trend is to design the tracking networks and pretrained them which aim to learn the target-specific features for each new sequence. And then the online tracking is followed using PF or classifiers [3, 19, 20]. Despite their notable performance, all these approaches are not designed towards real-time applications because of their time-consuming feature extraction and complex optimization details. In addition, they cannot end-to-end train and only tune the hyperparameters heuristically since feature extraction and tracking process are separate. Different from existing deep tracking frameworks, we formulate the extraction of high-level features as a one-layer convolution operation in the spatial domain. And the global handcrafted features are fused in a similar fully connection layer. This narrow deep learning structure allows feed-forward learning to capture robust appearance expression. Online adaptive tracking and updating mechanism brings optimal estimation for target location and scale selection.

3. The Tracking Framework

Our tracking algorithm includes constructing target model, online tracking, and model updating. The flowchart is expressed in Figure 1.

3.1. Joint Features Generation. Deep convolutional features can express the abstract information just like our brains, while different handcrafted features tend to deal well with some certain challenging tracking problems. Here we combine local convolutional features with color information as the appearance model of the tracking objects in order to deal with some challenging problems such as illumination variation and occlusion.

3.1.1. Local Convolutional Features. When there is heavy disturbance from complicated backgrounds, object shift will occur in current CNN tracking or correlation filter tracking. To avoid losing objects when involving in background information, our algorithm tries to design a group of local filters which not only consider the inner features of the target but also the background disturbance.

The target field is preprocessed into a fixed size and a set of overlapping patches are densely sampled inside it. To maintain good geometric and illumination invariance,

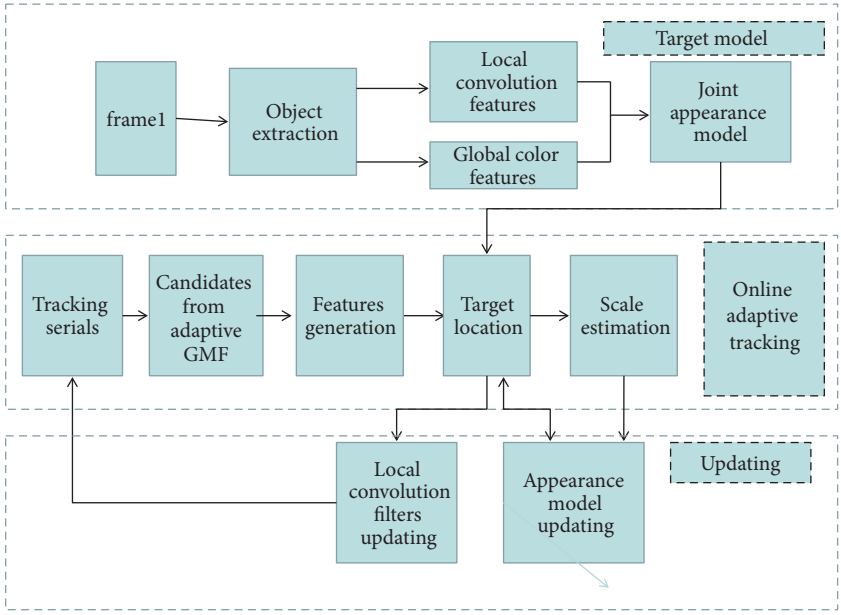

Figure 1: The algorithm flowchart.

we calculate the HOG features for each patch to express the local appearance and shape. After this, k-means algorithm is applied to select a foreground bank with n patches $\{P_1^O, P_2^O, ..., P_n^O\}$. Each patch is processed by subtracting the mean to get local contrast as the fixed foreground filters $\{F_1^O, F_2^O, ..., F_n^O\}$:

$$F_i^O = P_i^O - mean\left(P_i^O\right) \quad (i = 1, 2, \ldots, n) \tag{1}$$

Since the background context provides useful information to discriminate the target, here n background filters are generated from a group of samples. We choose m samples around the target in current frame, and a set of patches are selected in each sample. Just like foreground filters, we also calculate the HOG features and use k-means cluster to select n patches for each sample. After subtracting the mean value for each patch, a bank of filter $\{F_{i,1}^B, F_{i,2}^B, ..., F_{i,n}^B\}$ $(i = 1, 2, ..., m)$ is produced. Then we summarize all filters by weighted average to produce the background filters $\{F_1^B, ..., F_n^B\}$:

$$F_i^B = \frac{1}{m}\sum_{j=1}^{m} F_{j,i}^B \quad (i = 1, \ldots, n) \tag{2}$$

The final convolutional filters $\{F_1, ..., F_n\}$ are defined as the difference between the target filters and the background filters:

$$F_i = F_i^O - F_i^B \quad (i = 1, \ldots, n) \tag{3}$$

Then given the candidates $\mathrm{P} = \{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_\mathrm{m}\}$ in current input frame, the local convolutional feature map $D = \{D_1, D_2, ..., D_n\}$ for each candidate P_j is defined as

$$D_i = F_i \otimes P_j \quad \left(i = 1, \ldots, n;\ j = 1, \ldots, m\right) \tag{4}$$

In this way, the background information can be well suppressed, and shift brought by accumulative error gets alleviation. In addition, the feature maps produced by these filters are robust to noise introduced by appearance variations.

3.1.2. Global Color Features. Local convolutional features based on HOG captures the texture of the image while ignore color information. Color information has shown its advantages in the environments with shade or strong color contrast. So we use the global color histograms in HSV color space as the complementation of the local convolutional features.

According to the visual discrimination of human eyes, we quantify the three channels H, S, and V as

$$H = \begin{cases} 0 & H \in [316, 20] \\ 1 & H \in [21, 40] \\ 2 & H \in [41, 75] \\ 3 & H \in [76, 155] \\ 4 & H \in [156, 190] \\ 5 & H \in [191, 270] \\ 6 & H \in [271, 295] \\ 7 & H \in [296, 315] \end{cases} \tag{5}$$

$$S == \begin{cases} 0 & S \in [0, 0.2] \\ 1 & S \in (0.2, 0.7] \\ 2 & S \in (0.7, 1] \end{cases}$$

$$V == \begin{cases} 0 & V \in [0, 0.2] \\ 1 & V \in (0.2, 0.7] \\ 2 & V \in (0.7, 1] \end{cases}$$

Then the three parts are connected into 1D vector just as

$$G = HQ_SQ_V + SQ_V + V \tag{6}$$

Here, $Q_S = Q_V = 3$ are the quantization levels for S and V channels. The maximum value $G = 7 \times 9 + 2 \times 3 + 2 = 71$. Then the global color histogram H for a given image sample is defined as a vector $H = \{H(0), H(1), \ldots, H(71)\}$, and each element $H(i)$ is calculated by

$$H(i) = \frac{\sum_{t=1}^{M}\sum_{h=1}^{N}\delta(G(t,h) - i)}{M \times N} \tag{7}$$

where $M \times N$ is the number of the pixels in one sample. δ is Kronecker delta function. $G(t,h)$ is the fused value according to formula (6) at the location (t,h).

3.1.3. Joint Appearance Model. The convolutional feature maps D and color histogram features H encode the local structural texture information and global color information for the target. Thereby fusing them can bring a better representation to handle appearance variations and background disturbance. Here, we connect these two features and form a 1-D feature vector $A = [D, H]$, $A \in R^{1\times t}$ (t is the total number of the features) just like the fully connected layer in deep convolution network. In addition, to make the features robust to noises introduced by appearance variation, we normalize joint features A as the final appearance template:

$$A_j = \frac{A_j}{\sqrt{\sum_{i=1}^{t}(A_i)^2}} \quad (j = 1, \ldots, t) \tag{8}$$

Here, $A_i \in A$ denotes ith feature value. Hence the complex patch features preserve the geometric layouts of the useful parts and suppress confusing background in advance. The fused features give the full appearance description which can improve the tracking robustness.

3.2. Adaptive Tracking Algorithm

3.2.1. Position Estimation. Our tracking algorithm is formulated within an adaptive framework similar to particle filtering tracking. Different from current random sampling method, our algorithm introduce the speed information of consecutive frames to predict the candidate samples during tracking. We assume that m target states $\{(x_{it}, y_{it})\}_{i=1,\ldots,m}$ in current frame are modeled by two Gaussian distributions, and the dynamic model can be formulated as

$$\begin{aligned} x_{it} &= \frac{1}{\sqrt{2\pi}\sigma_{xt}}\exp\left(-\frac{(x_{t-1} - \mu_{xt})^2}{2\sigma_t^2}\right) \\ y_{it} &= \frac{1}{\sqrt{2\pi}\sigma_{yt}}\exp\left(-\frac{(y_{t-1} - \mu_{yt})^2}{2\sigma_{yt}^2}\right) \end{aligned} \tag{9}$$

Here, (x_{t-1}, y_{t-1}) is the target position in the previous frame. (μ_{xt}, μ_{yt}), $(\sigma_{xt}, \sigma_{yt})$ are separately mean and variance. They are determined by the motion information in the three previous frames.

$$\begin{aligned} \mu_{xt} &= \frac{(x_{t-1} + x_{t-2} + x_{t-3})}{3} \\ \mu_{yt} &= \frac{(y_{t-1} + y_{t-2} + y_{t-3})}{3} \end{aligned} \tag{10}$$

$$\begin{aligned} \sigma_{xt} &= sigx + \mu_{xt} \\ \sigma_{yt} &= sigy + \mu_{yt} \end{aligned} \tag{11}$$

where $sigx$, $sigy$ are constants, which can control the clustering degree of the candidate samples. Different from the traditional particle filter, the motion information gives the prediction of the candidates in the next frame. When there is partial even full occlusion, the samples can get more reasonable distribution.

Then the optimal state (x_t, y_t) is achieved by weighting all the predicting states:

$$(x_t, y_t) = \sum_{i=1}^{m} w_i \times (x_{it}, y_{it}) \tag{12}$$

The weighted values w_i $(i = 1, \ldots, m)$ play a key role in robust tracking. They are produced by the observation model T_i $(i = 1, \ldots, m)$:

$$w_i = \frac{T_i}{\sum_{i=1}^{m} T_i} \tag{13}$$

Here, T_i gives the distance between appearance representation A_t^i for the i_{th} candidate sample and the target template A_{t-1} at frame $t-1$. We hope the smaller the difference is, the bigger the contribution is. So the observation model T_i in this work is defined as

$$T_i = \frac{1}{\left(\left\|A_{t-1} - A_t^i\right\|_2^1 + \varphi\right)} \tag{14}$$

Here, φ is a minor positive constant.

With the proposed dynamic and observation model, the algorithm can keep tracking the target even if there is the overall occlusion and fast motion in the scene. In David sequence, the target walks behind one tree and occurs again. If only depending on the convolutional features, the tracking fails. The color information and motion information help to predict the target well (seen in Figure 2). In Figure 3, the target keeps rotating and moving fast; the visual tracker without color and motion cues during tracking causes the target lost.

3.2.2. Scale Estimation. Rotation or scale variation tends to appear when the object moves. To avoid involving extra background information or losing partial foreground, the tracking scale needs to be adjusted adaptively according to the tendency of scale change [21]. Supposing S_{t-1}, S_{t-2}, S_{t-3} are the sizes of the three former tracking results, we define the scale variation factor f as

$$f = \frac{S_{t-1}}{S_{t-2}} - \frac{S_{t-2}}{S_{t-3}} \tag{15}$$

Figure 2: The blue, green, and red show the tracking results of our model, fused features, and convolutional feature.

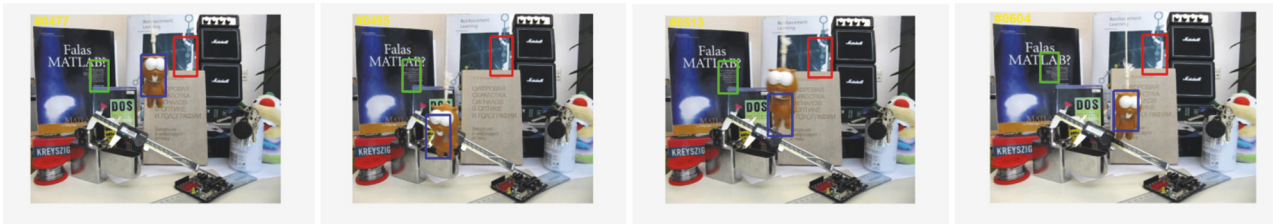

Figure 3: There is rotation and fast motion. The fused features and dynamic model bring continuous tracking.

If $f > 0$, the scale is regarded to be enlarged. We define an amplification pool $AM = \{am_1, ..., am_k\}$ for candidate scales. Then the new candidate scales $\mathbf{S} = \{S_t^i \mid i = 1, 2, ..., k\}$ in the current frame are calculated as

$$S_t^i = S_{t-1} \times am_i \quad am_i \in AM \tag{16}$$

To find the most proper scale, we calculate the distance according to formula (14) for each candidate with the size S_t^i. The final target scale S_t is chosen by

$$S_t = \arg\max_{S_t^i \in S} T^{S_t^i} \tag{17}$$

Else, the scale will be reduced. Similar to the scale amplification, we also set a reduction pool $DE = \{de_1, ..., de_k\}$ and choose the scale whose patch features are the most approximate to the model template.

3.3. Model Updating. Model updating is an important step in visual tracking. In the process of tracking, the object appearance often changes with the factors of scale, motion, rotation, or posture. Therefore, the appearance model needs to be updated over time to accommodate these changes for robust visual tracking. But overupdating is easy to result in shift and bring extra computation. Motivated by these intuitions, we consider the appearance model needs not to be updated when there's little change or large occlusion. For these two situations, we predefine two thresholds Th_1 and Th_2. Supposing $D_t = \|A_{t-1} - A'_{t-1}\|_1^2$ denotes the difference between current tracked target features A'_{t-1} and the target model A_{t-1}, then

if $D_t \prec Th_1$ or $D_t \succ Th_2$ the appearance template will not be updated.

Else, the model gets new appearance description A_t:

$$A_t = \rho \times A_{t-1} + (1 - \rho) \times A'_{t-1} \tag{18}$$

Here ρ is the weighted parameter and is set to 0.95. With the incremental update scheme, the appearance template is not only able to largely maintain the former appearance but also adapt to the target variations. And meanwhile the threshold and weighted parameter effectively control the updating degree. In this way, our method can alleviate the drift problem.

Due to illumination variation and target moving, the background keeps changing. To get correct appearance features for the predicting samples, we reproduce new background samples around current target location (x_t, y_t). The new convolution filters are recalculated according to formulas (2) and (3).

4. Experimental Results

4.1. Implementation Details. The proposed algorithm is tested on the OTB 100 dataset. The size and location of the target in the first frame are given by the ground-truth. In each frame, candidate samples are resized into 32×32 and each patch is 6×6. During online tracking, the number of the local filters is set to 80 and the adaptive Gaussian mixture filter produces 300 candidate samples. The constants $sigx, sigy$ are set as 2.5. The value for the scale amplification pool is $AM = \{1, 1.1, 1.2, 1.3, 1.4, 1.5\}$ and reduction pool $DE = \{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$.

4.2. Quantitative Comparisons. For quantitative evaluation, we compare our algorithm with 9 current trackers by reporting the results of one-pass evaluate (OPE) based on precision plots and success plots. The precision plots show the percentage of frames whose estimated location distance from the ground-truth is within a predefined threshold varying from 0 to 50 pixels. The success metric computes the intersection over union (IoU) and counts the number of successful frames whose IoU is larger than a given threshold (varying from 0 to 1).

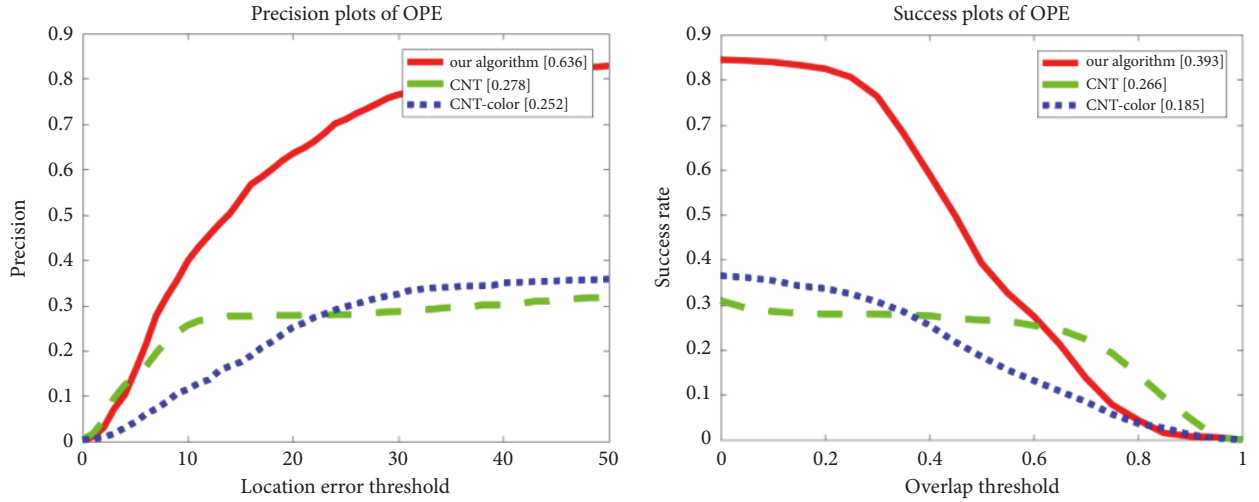

Figure 4: Comparisons between our algorithm and the two simplified versions.

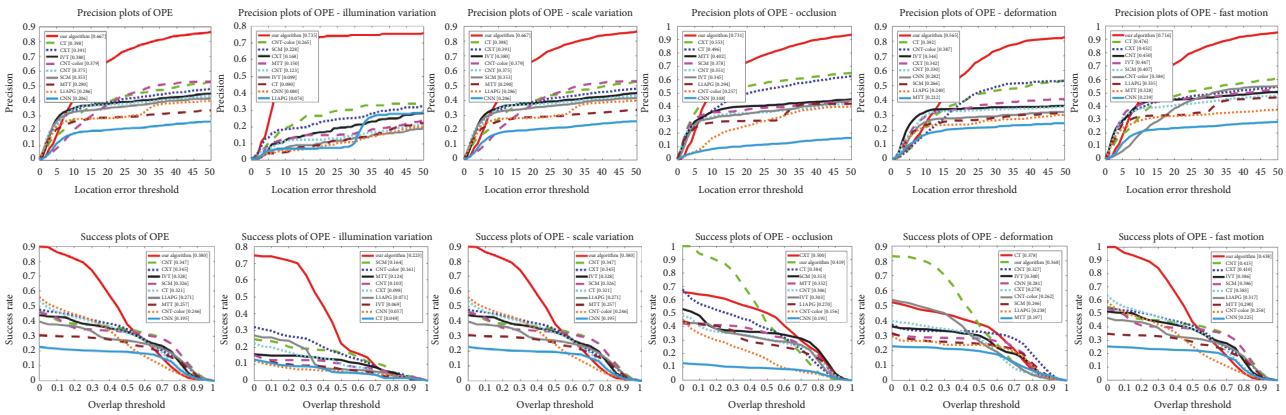

Figure 5: Comparison of different trackers with five attributes.

First of all, to highlight the contribution of local convolutional features, global handcrafted features, and motion information, we firstly compare our proposed algorithm with two simplified versions. They only depend on convolutional features (CNT) or on the convolutional and color fusion features (CNT-color). Five visual sequences (basketball, david3, human8, lemming, and biker) are tested with initialization from the ground-truth position in the first frame. These sequences include different challenging attributes such as fast motion, full occlusion, rotation, and illumination variation. The OPE curves in Figure 4 clearly show that our algorithm outperforms the other two versions by 32% ~53% in success plots and 56% ~60% in precision plots.

Then, to gain more insight about the proposed method, the proposed algorithm is compared with nine state-of-the-art visual trackers: CXT, IVT, CT, L1APG, CNN, MIT, CNT-color, CNT, and SCM. We use the publicly available source codes provided by the authors themselves with the same initialization and parameter settings to generate the comparative results. Here, success plots and precision plots of five different attributes are illustrated in Figure 5 which includes fast motion, occlusion, illumination variation, scale variation, and deformation. In the case of fast motion, the average precision plots and success plots show that our proposed method outperforms the other trackers by more than 40% and over 8%, respectively. Meanwhile, our method can outperform the CNN tracker which only depends on the deep convolutional features by a wide margin of more than 85% in terms of precision OPE and over 78% in success OPE for occlusion sequences. Though our method only ranks the second, it still maintains the best performance when the threshold is less than 0.5 shown in the success curve. In the challenging conditions of illumination and fast motion, our method gets the best results for both success and precision evaluations. This is largely due to the fact that moving information allows predicting the object direction well. Though several other trackers perform better in sequences with scale variation, the average statistics show that our method achieves competitive results (more than 39% and more than 4% in terms of precision rate and success rate, respectively) because of the fusion of local and global features.

4.3. Qualitative Comparisons. Figure 6(a) shows tracking results for the *David3* sequence. Performance on this

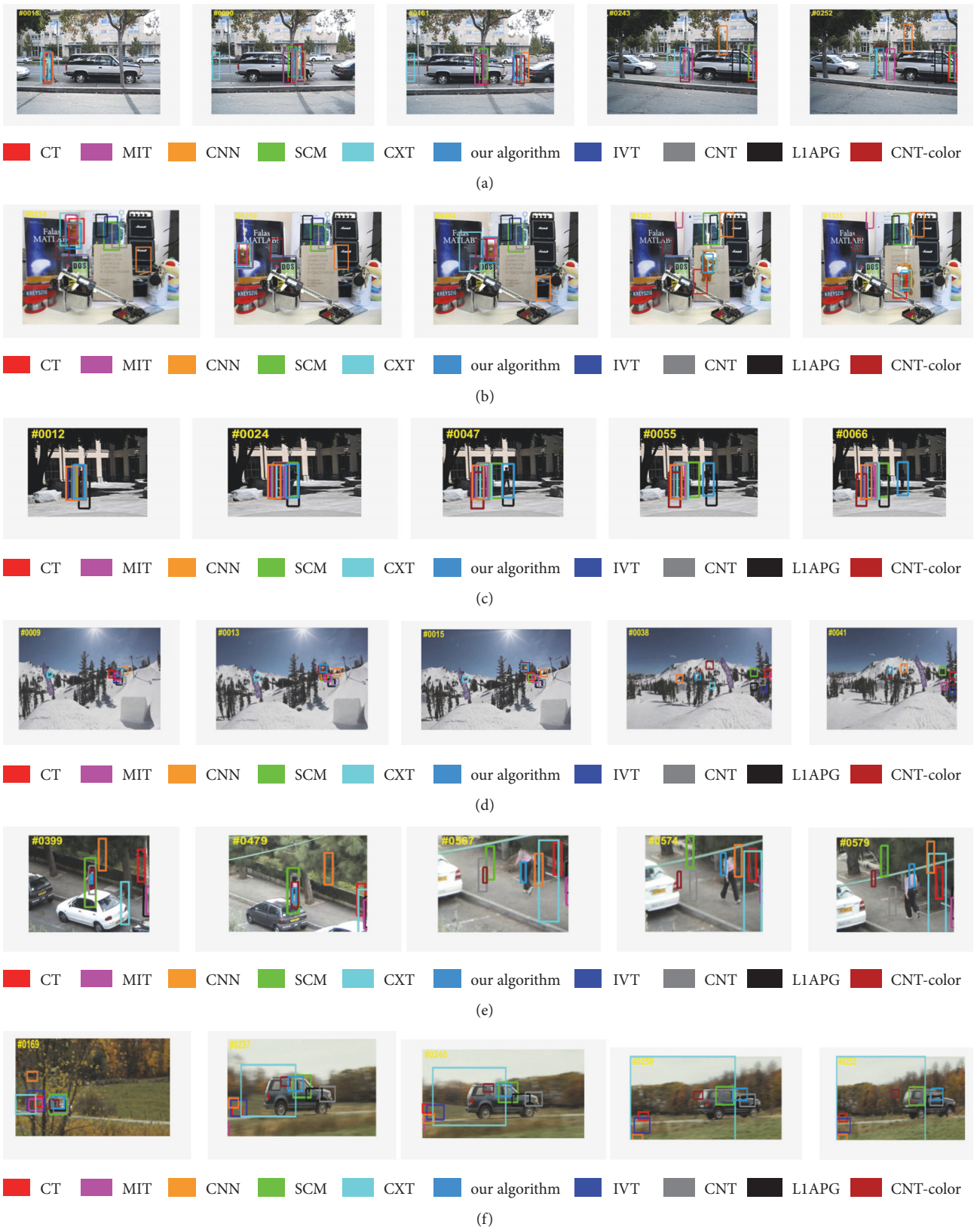

Figure 6: Qualitative results for selected sequence.

sequence exemplifies the robustness of the proposed algorithm to complete and partial occlusion. Only CNT-color and our algorithm are capable of tracking the target during the entire sequence. Other trackers experience drift at different instances: CXT at frame 15, MIT, L1APG, SCM, CT at frame 90, CNN at frame 161, and MIT at frame 243 because of partial occlusion.

Lemming sequence in Figure 6(b) includes full occlusion, fast motion, and scale variation. For the other methods, tracking drifts from the moving object while the proposed algorithm keeps tracking accurately because it exploits the motion tendency well.

In *Human8* sequence, the tracked object is subject to change in illumination and the background color is sometimes similar to the color of the man's backpack. CT, MIT, SCM, CXT, IVT, CNN, and CXT drift from the target continuously from frame 12 because of the similar color between the shade and the target. And meanwhile, L1APG and CNT cannot deal with the persistent change in illuminance and the disturbance of the background. They can only keep tracking until frame 55. Our method successfully tracks the object in the whole sequence.

Skiing sequence includes rotation in plane and fast motion. Tracking results at frames {9, 13, 15, 38, 41} for all 10 methods are shown. The different tracking methods are color-coded. CXT tracker starts to drift from the target at frame 7 and finally loses tracking. From frame 9, other trackers except our algorithm and CNT-color begin to drift and totally fail tracking at frame 15. When the target jumps quickly and rotates in the sky, only our method tracks the target successfully almost from frame 1 to frame 41.

In *Women* sequence, there are partial occlusion, motion blur and rotation. CNN, CT, CXT, MIT, IVT, and L1APG trackers lose the target with the occlusion occurring because of the background disturbance with the similar color (seen in frame 399 and frame 479). After the target passes the cars, fast walking brings a little blue effect at frame 567. At this moment, only our method can maintain good tracking. After that, the target turns back several times, our method can succeed tracking her with the help of fusion features and motion information.

The *CarScale* sequence shows a car keeps changing the scale and moving fast. When it passes a tree from frame 166, CXT, IVT, CT, MIT, and CNN begin to lose the target because of the occlusion and fail tracking at frame 169. After that, the scale changes continuously with the quick motion. Our method can not only keep tracking but also adjust the tracking speed with the moving car.

5. Conclusion

A novel visual tracking algorithm is proposed based on a simple online learning network. The fusion of global color features and local convolutional features shows robust tracking against shade and presence of confusing colors in the background. And meanwhile, the speed information directs the propagation of the particles and improves the adaptivity of PF. When there is total occlusion or quick motion, the proposed tracker can maintain robust tracking. The proposed algorithm achieves substantial performance gain over the existing state-of-the-art trackers.

Data Availability

(1) All the source images are from the TB-100 dataset which is publicly available online at http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html. (2) The other results including quantitative and qualitative comparisons can be available by emailing the authors at crystalhanlei@163.com.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This study is supported by the National Natural Science Foundation of China (Grant no. 61803199). Many thanks are due to the original authors for providing the TB-100 dataset that is publicly available online at http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html.

References

[1] S. Zhang, X. Lan, Y. Qi, and P. C. Yuen, "Robust visual tracking via basis matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 421–430, 2017.

[2] 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.

[3] X. Qian, L. Han, Y. Wang, and M. Ding, "Deep learning assisted robust visual tracking with adaptive particle filtering," *Signal Processing: Image Communication*, vol. 60, pp. 183–192, 2018.

[4] K. Zhang, Q. Liu, Y. Wu, and M. H. Yang, "Robust tracking via convolutional networks without training," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1779–1793, 2016.

[5] 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.

[6] J. Choi, H. J. Chang, S. Yun, T. Fischer, Y. Demiris, and J. Y. Choi, "Attentional correlation filter network for adaptive visual tracking," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR '17*, 6, p. 7, IEEE, July 2017.

[7] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017.

[8] Q. Liu, X. Lu, Z. He, C. Zhang, and W. Chen, "Deep convolutional neural networks for thermal infrared object tracking," *Knowledge-Based Systems*, vol. 134, pp. 189–198, 2017.

[9] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1779–1792, 2016.

[10] K. Hossain and C.-W. Lee, "Visual object tracking using particle filter," in *Proceedings of the International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 98–102, 2013.

[11] X. Li, S. Lan, J. Yue, and P. Xu, "Visual tracking based on adaptive background modeling and improved particle filter," in *Proceedings of the 2nd IEEE International Conference on Computer and Communications, ICCC '16*, pp. 469–473, October 2016.

[12] W. Ou, D. Yuan, D. Li, B. Liu, D. Xia, and W. Zeng, "Patch-based visual tracking with online representative sample selection," *Journal of Electronic Imaging*, vol. 26, no. 3, pp. 033006(1)–033006(2), 2017.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems 2012, NIPS '12*, pp. 1097–1105, December 2012.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '16*, pp. 770–778, July 2016.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the 29th Annual Conference on Neural Information Processing Systems, NIPS '15*, pp. 91–99, December 2015.

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

[17] Y. Qi, S. Zhang, L. Qin et al., "Hedged deep tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4303–4311, 2016.

[18] M. Danelljan, A. Robinson, F. Khan, and M. Felsberg, "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *Proceeding of the European Conference on Computer Vision (ECCV '16)*, 2016.

[19] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '16*, pp. 4293–4302, July 2016.

[20] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR '17*, pp. 6931–6939, July 2017.

[21] D. Yuan, X. Lu, D. Li, Y. Liang, and X. Zhang, "Particle filter re-detection for visual tracking via correlation filters," in *Multimedia Tools and Applications*, 2018.