2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Improved Correlation Filter Tracking with Hard Negative Mining

Chunguang Qie[1], Guanjun Guo[1], Yan Yan[1], Liming Zhang[2], Hanzi Wang[1,⋆]
[1] Fujian Key Laboratory of Sensing and Computing for Smart City,
School of Information Science and Technology Xiamen University, Xiamen, China
[2] Faculty of Science and Technology University of Macau, Macau, China
Email: {qie_chunguang, ggj05}@qq.com, {yanyan, hanzi.wang}@xmu.edu.cn, lmzhang@umac.mo

***Abstract*—Recently, the correlation filter based trackers have achieved very good tracking performance. However, due to the boundary effects of the circulant matrix and the usage of cosine window, the lack of effective negative samples becomes a challenging problem for the correlation filter based trackers. This problem may cause overfitting so that these trackers become very sensitive to deformation and occlusion. In this paper, we propose a novel object tracker (*i.e.,* STAPLE_HNM), which can effectively select hard negative samples and assign adaptive weights to these samples to train the correlation filter. Experimental results demonstrate that the proposed STAPLE_HNM tracker effectively improves the performance of the baseline STAPLE_CA tracker on the OTB-50 and OTB-100 datasets. Moreover, the proposed STAPLE_HNM tracker also achieves superior performance among several state-of-the-art trackers.**

I. Introduction

Visual tracking serves as one of the most important components in a variety of computer vision systems, such as video surveillance, robots, and driverless cars. However, as it usually needs to handle all the variability caused by illuminations, occlusions, and erratic moving objects, it remains one of the challenging problems in computer vision.

In recent years, the correlation filter (CF) has been widely employed in the tracking algorithms due to its fast speed and high accuracy. In general, the CF-based trackers first learn the correlation filters by employing the discrete Fourier transform (DFT) and then detect the target object using the learned CF in the following consecutive frames. Specifically, the CF-based trackers construct a circulant matrix of training samples and then solve the optimization problem in the frequency domain. For each frame, these trackers detect the target object based on the response map, which is obtained by applying the learned CF to the regions of interest.

There are several problems that affect the efficiency of the CF-based trackers. One of the problems is that the circular shifts in the circulant matrix cause the boundary effects. The boundary effects limit the searching region of the CF-based trackers, thus leading to drifting in some challenging scenes, (*e.g.,* fast motion and occlusion). A common strategy to mitigate the boundary effects is to apply a cosine window to the region of interest. However, the cosine window suppresses the boundary region of the image patch and thus the quality of samples, especially the quality of negative samples, is drastically degraded. Thus, the lack of meaningful negative samples limits the capability of the CF-based trackers to cope with occlusion and deformation.

Some significant improvements have been proposed within the CF framework in the literature [1]–[11]. For example, the KCF tracker [3] demonstrates that the kernel tricks could be integrated into the CF-based trackers to improve the tracking performance. The DSST tracker [12] takes advantage of the adaptive scale to further improve the tracking performance. However, both of the trackers suffer from the boundary effects. The SRDCF tracker [4] improves the CF-based trackers by introducing the spatial regularization strategy to mitigate the boundary effects. However, solving the optimization problem in the SRDCF tracker is time-consuming. Recently, some context-aware [9] and background-aware [8] CF-based trackers have shown obvious improvements on the tracking performance. These trackers uniformly select negative samples around the target object. However, they have two major problems: (1) Some negative samples may be similar to each other, which causes the redundancy of negative samples in the training stage; (2) The negative samples are selected by only considering their spatial distances to the target object, which leads to the problem that some selected negative samples are not informative.

In this paper, we propose a new hard negative mining algorithm to improve the robustness of CF-based tracker. We summarize our work as follows:

1) We propose a novel strategy of selecting hard negative samples for the CF-based trackers, which can effectively reduce the redundancy of the hard negative samples.
2) The weights of the hard negative samples are adaptively computed and used in the training stage. We assign large weight values to the hard negative samples that are similar to the target object, by which, the discriminative ability of CF can be improved.

The proposed hard negative mining algorithm is shown in Fig. 1. In this paper, we integrate the proposed hard negative mining algorithm into the recently developed STAPLE_CA [8] tracker, and we call the proposed tracker STAPLE_HNM (*i.e.,* STAPLE with hard negative mining).

⋆Corresponding author

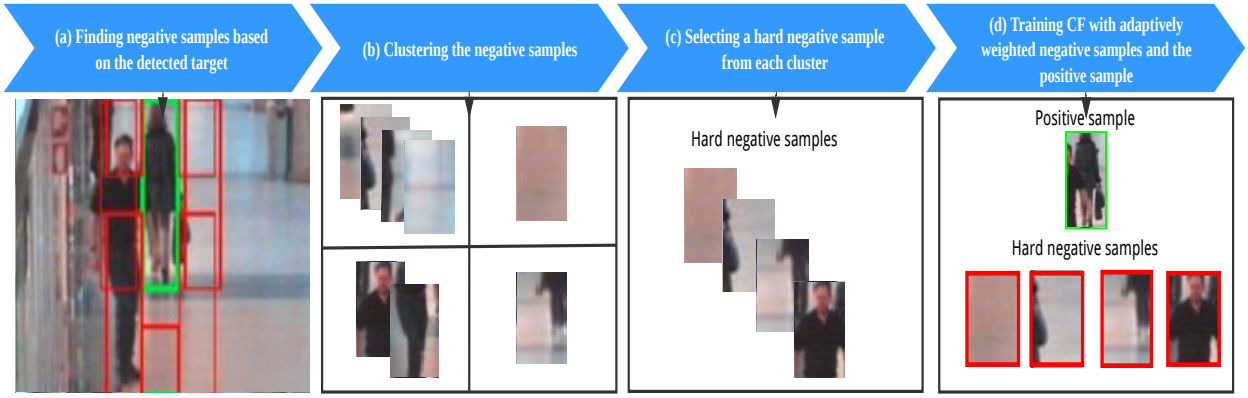


Fig. 1. The process of the proposed hard negative mining algorithm. In each new frame, (a) the proposed algorithm selects the negative samples (*i.e.,* the red rectangles) around the detected target object (*i.e.,* the green rectangle); (b) The negative samples are clustered into different clusters; (c) The hard negative samples are selected from these clusters; (d) The adaptive weight value for each hard negative sample is computed and the proposed STAPLE_HNM tracker trains the CF with the positive sample and the weighted hard negative samples.

II. The Proposed Algorithm

The improvements of the proposed algorithm STAPLE_HNM tracker are twofold: One is the strategy of selecting hard negative samples and the other is the way of training the CF with adaptively weighted hard negative samples. We begin with a brief review of the KCF tracker [3].

A. The Review of the KCF tracker

The training objective of the KCF tracker [3] is to find a prediction function $f(\boldsymbol{x})$ that minimizes the squared error over a square matrix $\boldsymbol{X}$ and the corresponding Gaussian shaped regression targets $\boldsymbol{y}$, *i.e.,*

$$\min_{\beta} ||\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}||^2 + \lambda||\boldsymbol{\beta}||^2, \tag{1}$$

where the rows in $\boldsymbol{X}$ are the circulant shifted samples of $\boldsymbol{x}$, $\boldsymbol{\beta}$ is the parameter of the prediction function and λ denotes the regularization parameter, which is used to prevent overfitting.

By taking advantage of the properties of circulant matrix [13], the parameter $\boldsymbol{\beta}$ can be efficiently estimated in the frequency domain:

$$\hat{\boldsymbol{\beta}} = \frac{\hat{\boldsymbol{x}^*} \odot \hat{\boldsymbol{y}}}{\hat{\boldsymbol{x}^*} \odot \hat{\boldsymbol{x}} + \lambda}, \tag{2}$$

where each variable with a hat denotes the DFT of that variable, $*$ denotes the complex-conjugate operator and $\odot$ denotes element-wise multiplication.

In order to detect the location of the target object in a new frame, the regression results in the frequency domain are obtained as the element-wise product of the parameter $\boldsymbol{\beta}$ and the image patch $\boldsymbol{x'}$:

$$\hat{f}(\boldsymbol{x'}) = \hat{\boldsymbol{\beta}} \odot \hat{\boldsymbol{x'}}. \tag{3}$$

Then we employ the inverse DFT to convert $\hat{f}(\boldsymbol{x'})$ to the spatial domain, *i.e.,* $f(\boldsymbol{x'})$. At last, the location of the target object is obtained by finding the maximum value in $f(\boldsymbol{x'})$.

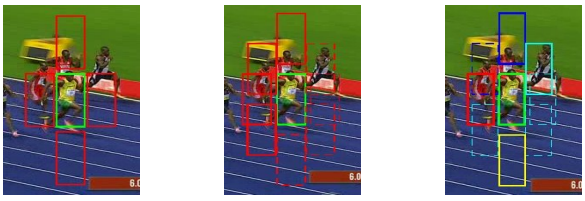
We define

$$r(\boldsymbol{x'}) = max(f(\boldsymbol{x'})), \tag{4}$$

which can be used to measure the similarity between $\boldsymbol{x'}$ and the target object.

B. Strategy of Selecting Hard Negative Samples

Firstly, we show different strategies of selecting hard negative samples in Fig. 2. To obtain negative samples, a simple strategy is to uniformly select the image patches in the background around the target object (as shown in Fig. 2 (a)). However, as we mention in Section I, the uniform selection strategy may lead to meaningless negative samples, and such a strategy may limit the performance of a tracker. Another straightforward strategy of selecting hard negative samples is to apply the CF learned from the previous frames to a set of negative samples $\{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_n\}$, and select the negative samples with the top m response values as the hard negative samples $\{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_m\}$. However, some of the selected hard negative samples are similar to each other in appearance, as shown in Fig. 2 (b). Therefore, there exists great redundancy in $\{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_m\}$, which may reduce the effectiveness of the training stage in the tracker.



(a) uniform strategy (b) top-m strategy (c) proposed strategy

Fig. 2. Different strategies of selecting hard negative samples. The solid green rectangle denotes the positive sample and the other solid rectangles denote the selected hard negative samples. The dash rectangles denote the candidate of negative samples. The different colors of rectangles denote different clusters of negative samples.

In this paper, we propose a new strategy to alleviate the redundancy of negative samples by clustering these samples based on their feature similarity and spatial distance in the image plane. Then we select a negative sample with the

largest response value in each cluster as the hard negative sample. In our case, we employ the K-means algorithm for clustering. However, how to measure the distance between negative samples has significant influence on the clustering results. While the maximum possible distance between two feature vectors (*e.g., HOG [14], RGB or grayscale*) is limited, the spatial distance in the image plane depends on the size of image patch. Therefore, it is not effective to measure the Euclidean distance between negative samples by a $(S+2)$-D vector without normalizing the distance between the 2-D spatial coordinate, where S denotes the length of feature vectors. To address this problem, we propose a new distance formula D_{ij} to measure the distance between the i-th negative sample and the j-th negative sample as follows:

$$\begin{aligned} d_{ij}^{f} &= \sqrt{\sum_{s=1}^{S}(a_{is}-a_{js})^2}, \\ d_{ij}^{l} &= \sqrt{(p_i-p_j)^2+(q_i-q_j)^2}, \\ D_{ij} &= d_{ij}^{f} + \frac{\rho}{\sqrt{wh}} d_{i,j}^{l}, \end{aligned} \tag{5}$$

where a_{is} and a_{js} respectively denote the s-th entries of the i-th and j-th feature vectors of two negative samples $\boldsymbol{a_i}$ and $\boldsymbol{a_j}$, p_i and q_i respectively denote the horizontal and vertical coordinates of the i-th sample, w and h respectively denote the width and height of the sample, and ρ is the parameter used to control the importance of the spatial distance in D_{ij}.

Then we use Eq. (5) as the distance metric for the K-means algorithm to generate m clusters. At last, we select m hard negative samples from these clusters. Specifically, in each cluster, we select a negative sample with the maximal response value in the cluster as the hard negative sample. Therefore, we can obtain a set of hard negative samples $\{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_m\}$ from the m clusters. From Fig. 2 (c), we can observe that the proposed strategy not only reduces the redundancy of negative samples, but also selects high-quality hard negative samples.

C. Training CF with Adaptively Weighted Hard Negative Samples

Now we can train the CF using the target sample $\boldsymbol{x}$ and the hard negative samples $\{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_m\}$. Based on [8], we propose a novel objective function, which assigns an adaptive weight to each hard negative sample:

$$\min_{\beta} ||\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}||^2 + \lambda||\boldsymbol{\beta}||^2 + \sum_{i}^{m} u_i||\boldsymbol{H}_i\boldsymbol{\beta}||^2, \tag{6}$$

where u_i denotes the adaptive weight of the hard negative sample $\boldsymbol{h}_i$, $\boldsymbol{H}_i$ denotes the square matrix, which contains all the circulant shifted samples of $\boldsymbol{h}_i$, and m denotes the number of hard negative samples.

Firstly, we define the weight u_i of the hard negative sample $\boldsymbol{h}_i$ as

$$u_i = \frac{r(\boldsymbol{h}_i)}{\sum_{j=1}^{m} r(\boldsymbol{h}_j)}, \tag{7}$$

where $r(\boldsymbol{h}_i)$ and $r(\boldsymbol{h}_j)$, defined in Section II-A, are respectively the response values of two hard negative sample $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$. A normalized weight is computed for each hard negative sample by using Eq. (7). Specifically, when a negative sample obtains a high response $r(\boldsymbol{h}_i)$, its corresponding weight becomes large. Thus, the CF is trained to focus more on the hard negative samples, which are more similar to the target object.

As mentioned by [8], we can stack the hard negative samples under the target sample to construct a new data matrix $\boldsymbol{P}$, and concatenate $\boldsymbol{y}$ with zeros to form a new regression target $\boldsymbol{Y}$. Then the the objective function in Eq. (6) can be rewritten as:

$$\min_{\beta} ||\boldsymbol{U}\boldsymbol{P}\boldsymbol{\beta}-\boldsymbol{Y}||^2 + \lambda||\boldsymbol{\beta}||^2, \tag{8}$$

where

$$\boldsymbol{U} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & u_1 & 0 & \dots & 0 \\ 0 & 0 & u_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & u_m \end{bmatrix}, \boldsymbol{P} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \vdots \\ \boldsymbol{H}_m \end{bmatrix} \text{ and } \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix}.$$

Because $\boldsymbol{U}$ can be considered as the weight matrix of the samples in $\boldsymbol{P}$, Eq. (8) is convex and its closed-form solution can be written as:

$$\boldsymbol{\beta} = (\boldsymbol{U}\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{U}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{P}^T\boldsymbol{U}^T\boldsymbol{Y}, \tag{9}$$

where $\boldsymbol{I}$ denotes the identity matrix.

Based on [8], the estimate of $\boldsymbol{\beta}$ in the frequency domain is written as:

$$\hat{\boldsymbol{\beta}} = \frac{\hat{\boldsymbol{x}^*} \odot \hat{\boldsymbol{y}}}{\hat{\boldsymbol{x}^*} \odot \hat{\boldsymbol{x}} + \lambda + \sum_{i=1}^{m} u_i^2 \hat{\boldsymbol{h}_i^*} \odot \hat{\boldsymbol{h}_i}}. \tag{10}$$

Once we estimate the value of $\boldsymbol{\beta}$, we can use Eq. (3) to estimate $\hat{f}(\boldsymbol{x'})$ and locate the target object by finding the maximum value in the inverse DFT of $\hat{f}(\boldsymbol{x'})$.

D. Implementation Details

To demonstrate the effectiveness of the strategies of selecting hard negative samples and using the adaptive weights for hard negative samples, we integrate these two components into the STAPLE_CA tracker [8], which originally selects negative samples uniformly around the target object and uses the same weight value for all the negative samples.

In the proposed STAPLE_HNM tracker, we set the number of candidate negative samples $n = 8$ and the number of hard negative samples $m = 4$. We set the weight parameter ρ in Eq. (5) to be 1. All the candidate negative samples are uniformly distributed in the regions around the target object. In the clustering stage, the center of each cluster is uniformly initialized around the target object, as shown in Fig. 2(a). In each frame, we set the number of iterations in the K-means algorithm to be 2. Besides, the proposed tracker employs the HOG feature descriptor with the cell size 4.

III. Experiments

In this section, we first verify the effectiveness of each component of the proposed algorithm by comparing several variants of the proposed STAPLE_HNM tracker. Then we demonstrate the outstanding performance of the STAPLE_HNM tracker by comparing it with several state-of-the-art trackers.

The experimental results on the OTB-100 [15] and OTB-50 [16] datasets are reported. The OTB-100 dataset and OTB-50 dataset contain 100 test video sequences and 50 test video sequences, respectively. We employ the one-pass evaluation (OPE) to evaluate the competing trackers, *i.e.*, initialize the tracker in the first frame and let it track the target until the end of the sequence. All the competing trackers are evaluated according to two metrics: precision and success rate. The precision measures the center location error, which computes the average Euclidean distance between the center locations of the tracked targets and the groundtruth positions for the target for all the frames. The success rate measures the intersection over the union of the predicted bounding box and the groundtruth bounding box. Besides, the test sequences on the OTB-100 dataset involve 11 challenging attributes, such as illumination changes, motion blur, scale variation, deformation, *etc.*.

A. Components Analysis

In this section, we evaluate the tracking performance obtained by 4 variants of STAPLE_HNM: (1) STAPLE_HNM-C$^-$W$^-$, which employs top-m strategy (as shown in Fig. 2 (b)) and it does not use the adaptive weights for hard negative samples; (2) STAPLE_HNM-W$^-$, which uses the proposed strategy of selecting hard negative samples, but it does not assign adaptive weights to hard negative samples; (3) STAPLE_HNM, which employs both the proposed strategy of selecting hard negative samples and the proposed adaptive weighting strategy; (4) STAPLE_CA, which is the baseline tracker without hard negative mining.

TABLE I
Precision and success rate obtained by 4 variants of STAPLE_HNM on the OTB-100 and OTB-50 datasets. The first, second and third best scores are highlighted in red, blue and green colors, respectively.

Tracker	OTB-50		OTB-100	
	Precision	Success	Precision	Success
STAPLE_HNM	85.32	63.20	82.81	60.99
STAPLE_HNM-W$^-$	84.93	62.75	82.01	60.42
STAPLE_HNM-C$^-$W$^-$	84.05	62.22	81.23	60.12
STAPLE_CA	83.26	62.14	80.95	60.01

The precision and success rate obtained by the 4 variants on both OTB-50 and OTB-100 are shown in Table I. From Table I, we have the following conclusions: (1) Since the CF is trained by using more effective negative samples, the trackers, which select hard negative samples (*i.e.,* STAPLE_HNM, STAPLE_HNM-W$^-$), outperform the baseline tracker STAPLE-CA. (2) By comparing the performance of STAPLE_HNM-W$^-$ and STAPLE_HNM-C$^-$W$^-$, we can observe that the proposed strategy of selecting hard negative samples can effectively improve the tracking performance of STAPLE_HNM-W$^-$. This is because that selecting hard negative samples from clusters effectively alleviates the redundancy of all the hard negative samples in both the feature space and spatial space. (3) According to the the precision and success rates obtained by STAPLE_HNM and STAPLE_HNM-W$^-$, the proposed adaptive weighting strategy can further improve the tracking performance. Therefore, each of the two components in the proposed STAPLE_HNM tracker is helpful to improve the final tracking performance.

B. Comparison with State-of-the-art Trackers

In this section, we perform comprehensive comparison with 9 state-of-the-art trackers: SRDCF [4], STAPLE_CA [8], CFNet [7], STAPLE [17], SiameseFC [18], SAMF [6], DSST [12], KCF [3] and CSK [19]. Except for the SiameseFC, which uses the cross-correlation for tracking, all the other trackers are CF-based. Specifically, CFNet and SiameseFC employ the convolutional neural network (CNN) to improve their performance.

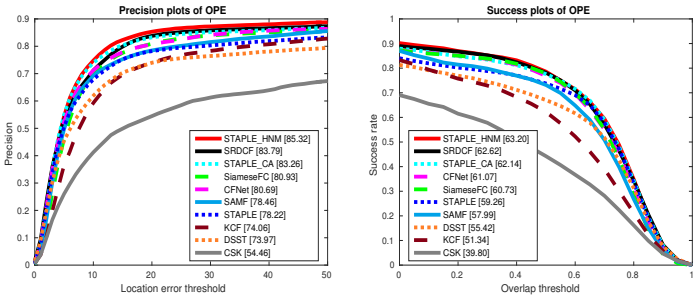


Fig. 3. Precision plots and success plots obtained by the 10 competing trackers on the OTB-50 dataset

Results on the OTB-50 Dataset. Both the precision plots and the success rate plots obtained on the OTB-50 dataset are illustrated in Fig. 3. As shown in Fig. 3, the proposed STAPLE_HNM tracker with the precision of 85.32% and success rate of 63.20% respectively, outperforms all of the 9 competing trackers. Compared with the STAPLE_CA tracker, the proposed STAPLE_HNM tracker outperforms STAPLE_CA by 2.06% and 1.06% on precision and success rate, respectively. The results obtained by STAPLE_HNM and STAPLE_CA demonstrate that using hard negative samples rather than uniformly selecting negative samples to train the tracker can effectively reduce drifting and obtain tight bounding boxes. Although CFNet employs CNN to extract robust features, the proposed STAPLE_HNM achieves better performance than it because that CFNet suffers from the problem of the circulant effects, which leads to the lack of effective negative samples.
Results on the OTB-100 Dataset. Fig. 4 illustrates both the precision and success rate plots obtained by all the 10 competing trackers on the OTB-100 dataset. As shown in Fig. 4, the proposed STAPLE_HNM tracker, with the precision of 82.81% and success rate of 60.99% respectively, still outperforms all the other competing trackers, which demonstrates the state-of-the-art performance of the proposed STAPLE_HNM

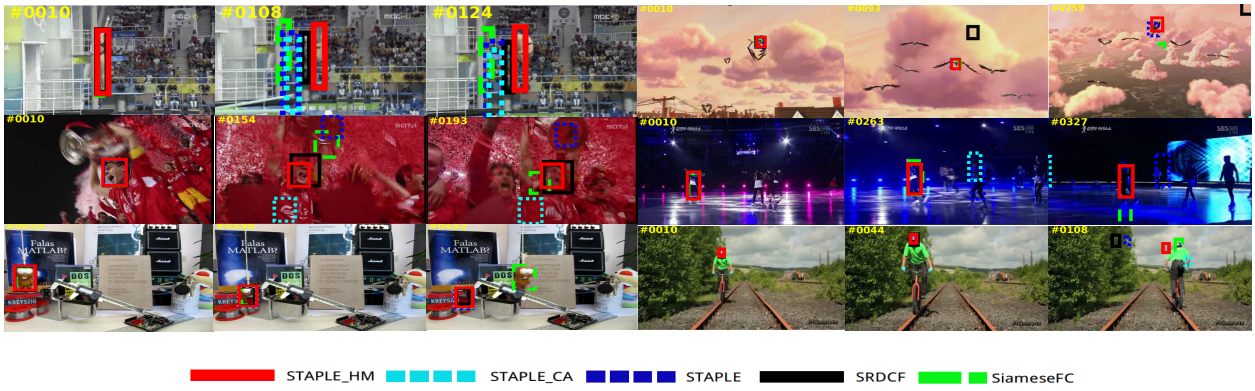


Fig. 5. Qualitative comparison of the proposed tracker with several state-of-the-art trackers on the *diving*, *bird1*, *soccer*, *skating1*, *lemming* and *biker* test sequences. In the first two rows, the proposed tracker provides consistent results in challenging scenarios, such as deformation, motion blur, and illumination variation. The last row shows failure cases in the *lemming* and *biker* test sequences.

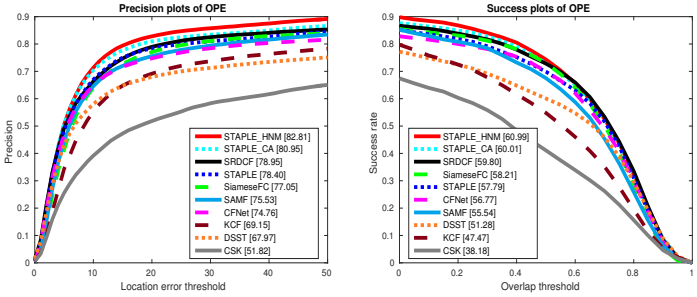


Fig. 4. Precision plots and success plots obtained by the 10 competing trackers on the OTB-100 dataset

tracker. Specifically, compared with the baseline tracker STAPLE_CA, the proposed STAPLE_HNM tracker outperforms it by 1.86% and 0.98% on precision and success rate, respectively. Compared with SAMF, the proposed STAPLE_HNM outperforms it by 6.86% and 5.37% on precision and success rate, respectively. Moreover, the proposed STAPLE_HNM only uses the HOG feature descriptor instead of the integrated HOG and ColorName features, which are used in SAMF. This suggests that the tracking performance of the proposed STAPLE_HNM tracker may be further improved by employing more robust feature descriptors.

Qualitative Evaluation. Fig. 5 shows the qualitative comparison of the proposed tracker with several existing trackers on some challenging video sequences. From the first two rows of Fig. 5, we can see STAPLE_HNM achieves favorable results under several challenging scenarios, such as deformation, motion blur, and illumination variation. However, we also observe that there are some failure cases of the proposed tracker and we show them in the third row of Fig. 5. Most of the failures are caused by the long-term occlusion and fast motion, which lead to drifting. Due to the limited searching region, it is hard for the proposed STAPLE_HNM tracker to recover from drifting.

Tracking Speed. We evaluate the average speed of the proposed tracker on the OTB-100 dataset. It is implemented in Matlab on a workstation equipped with a 3.60GHz CPU. The proposed STAPLE_HNM tracker runs at 24 FPS. The best tracking speed is achieved by CSK (479.3 FPS), followed by KCF (170 FPS). Although these trackers achieve higher speeds than the proposed tracker, the performances are much lower than the proposed STAPLE_HNM tracker (*i.e.,* the proposed tracker outperforms CSK more than 20% and KCF more than 10% on both precision and success rate on OTB-100 dataset, respectively). The baseline tracker STAPLE_CA runs at 44.5 FPS. Due to the usage of the proposed strategies of selecting hard negative samples and computing the adaptive weights, the proposed STAPLE_HNM tracker runs slower than STAPLE_CA, but STAPLE_HNM still satisfies the real-time requirements.

Attribute-Based Analysis. We evaluate the tracking performance of all the competing trackers under 11 challenging attributes annotated in the OTB-100 dataset, and the results are shown in Fig. 6. From Fig. 6, we have the following conclusions: (1) The proposed tracker achieves the best tracking performance for 10 out of 11 attributes (*i.e.*, on all the attributes except for low resolution). This is because that the proposed tracker trains the CF with effective hard negative samples, which enhances the robustness of the tracker. For the low resolution attribute, the proposed tracker obtains the lower tracking precision than the SiameseFC [18] and CFNet [7] trackers, which take advantage of the robust CNN features. (2) Compared with the baseline tracker STAPLE_CA, the proposed tracker achieves better tracking precision for all the 11 challenging attributes. This is because that the proposed tracker selects more informative hard negative samples and the CF is trained more effectively due to the usage of these hard negative samples and the adaptive weights for the hard negative samples.

IV. Conclusion

In this paper, we have proposed an improved CF-based tracker, called STAPLE_HNM, which can effectively select hard negative samples and reduce the redundancy within hard negative samples. Besides, the proposed STAPLE_HNM tracker trains the CF by using adaptive weights for different hard negative samples, which can further improve the

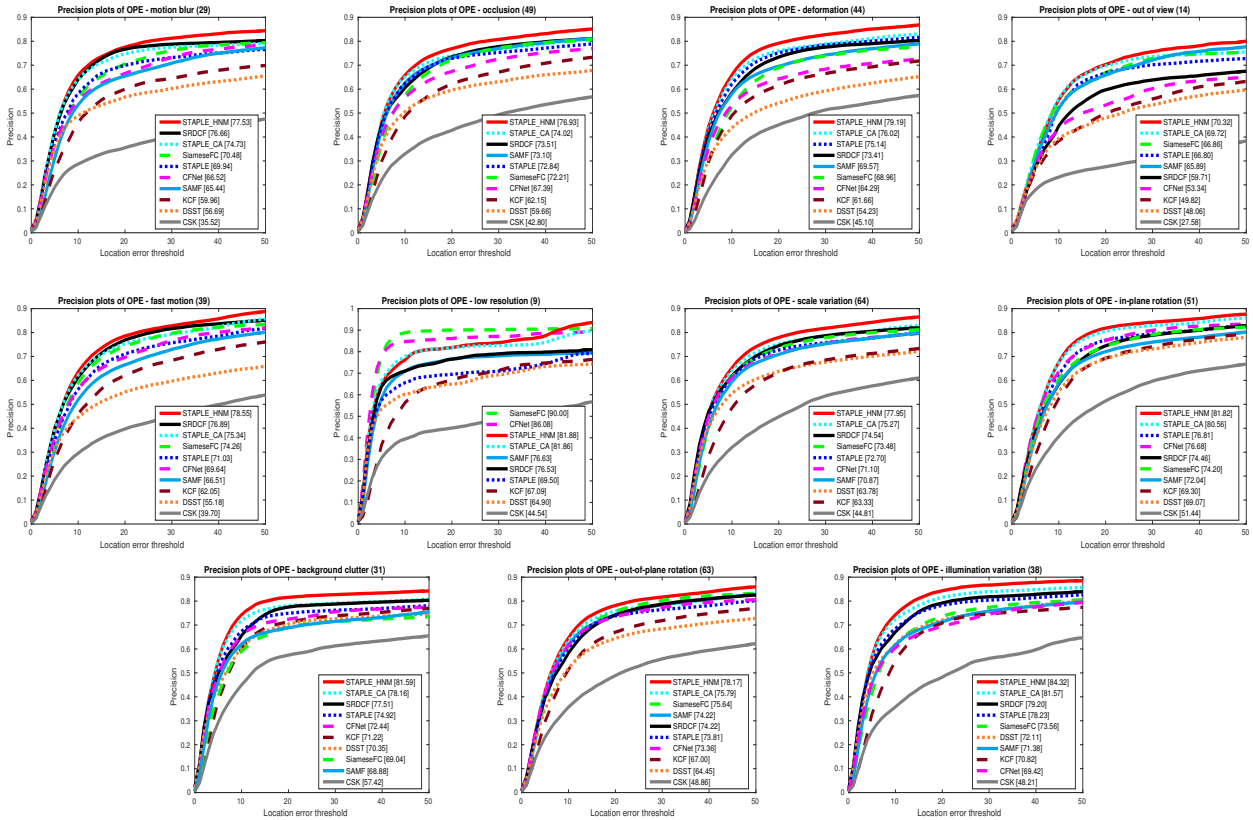


Fig. 6. Precision plots obtained by the 10 competing trackers over the 11 attributes annotated in the OTB-100 dataset. The values in the title are the number of sequences under that attribute.

tracking performance. We conduct comprehensive experiments to demonstrate the effectiveness of each component of the proposed STAPLE_HNM tracker. Furthermore, compared with several state-of-the-art trackers, the proposed tracker achieved the best performance on both the OTB-50 and OTB-100 datasets.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grants U1605252, 61472334 and 61571379, by the Natural Science Foundation of Fujian Province of China under Grant 2017J01127, by the National Key R&D Program of China under Grant 2017YFB1302400, and by the UM Multi-year Research under Grant MYRG2017-00218-FST.

REFERENCES

[1] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking." in *CVPR*, 2015, pp. 5388–5396.

[2] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters." in *CVPR*, 2010, pp. 2544–2550.

[3] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," IEEE *Trans. PAMI*, vol. 37, no. 3, pp. 583–596, 2014.

[4] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *ICCV*, 2015, pp. 4310–4318.

[5] M. Danelljan, F. S. Khan, M. Felsberg, and J. v. d. Weijer, "Adaptive color attributes for real-time visual tracking," in *CVPR*, 2014, pp. 1090–1097.

[6] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *ECCV Workshops*, 2014, pp. 254–265.

[7] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *CVPR*, 2017, pp. 5000–5008.

[8] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *CVPR*, 2017, pp. 1387–1395.

[9] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *ICCV*, 2017.

[10] L. Si, Z. Tianzhu, C. Xiaochun, and X. Changsheng, "Structural correlation filter for robust visual tracking," in *CVPR*, 2016, pp. 4312 – 4320.

[11] B. Adel, M. Matthias, and G. Bernard, "Target response adaptation for correlation filter tracking," in *ECCV*, 2016, pp. 419–433.

[12] M. Danelljan, G. Hger, F. Shahbaz Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*, 2014.

[13] R. M. Gray, "Toeplitz and circulant matrices: A review." Now Publishers Inc, 2006.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, pp. 886–893.

[15] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," IEEE *Trans. PAMI*, vol. 37, no. 9, pp. 1834–1848, 2015.

[16] W. Yi, L. Jongwoo, and Y. Ming-Hsuan, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.

[17] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *CVPR*, 2016, pp. 1401 – 1409.

[18] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV 2016 Workshops*, 2016, pp. 850–865.

[19] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *ECCV*, 2012, pp. 702–715.