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

Learning Collaborative Model for Visual Tracking

Ding Ma*, Wei Bu†, Yuehua Cui‡, Yuying Xie§ and Xiangqian Wu*
*School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China 150001
Email: xqwu@hit.edu.cn
†Department of New Media Technologies and Arts
Harbin Institute of Technology, Harbin, China 150001
‡Department of Statistics and Probability
Michigan State University, East Lansing, USA 48824
§Department of Computational Mathematics, Science and Engineering
Michigan State University, East Lansing, USA 48824

***Abstract*—This paper proposes a robust visual tracking method by designing a collaborative model. The collaborative model employs a two-stage tracker and a HOG-based detector, which exploits both holistic and local information of the target. The two-stage tracker learns a linear classifier from the patches of original images and the HOG-based detector trains a linear discriminant analysis classifier with the object exemplar. Finally, a result decision making strategy is developed by considering both the original template and the appearance variations, making the tracker and the detector collaborate with each other. The proposed method has been evaluated on OTB-50, OTB-100 and Temple-Color datasets, and results demonstrate that the proposed method is able to effectively address the challenging cases such as scale variation and out-of-view and gets better performance than the state-of-the-art trackers.**

I. Introduction

In the task of visual tracking, an object is identified in the first video frame and should be tracked in subsequent frames. And the tracking algorithm should be robust enough to handle the various appearance variations of the taret. Despite great progress in recent decades, visual tracking remains a challenging task due to illumination change, heavy occlusion, and complex background (see Fig. 1).

A visual tracking method can be seen as a system that consists of five components [1]: the feature extractor, the appearance model, the motion model, the model updater, and the ensemble post-processor. Given the first frame of a video, one or more suitable features are extracted to construct the appearance model of the target. The appearance model provides the confidence of a given candidate being the target. Based on the candidate with the highest confidence from the previous frame, numerous candidates of the target are generated from the motion model. Depending on the model updater, the appearance model is gradually changed to adapt the variation of the target. However, even with a little change of the parameters, the performance of a single tracker would dramatically change. Ensemble of multi-post-processor is available to overcome this boundedness. In this paper, we not only design a robust appearance model which plays an important role in a tracker, but also take into account the efficiency of the other four components of the tracker.

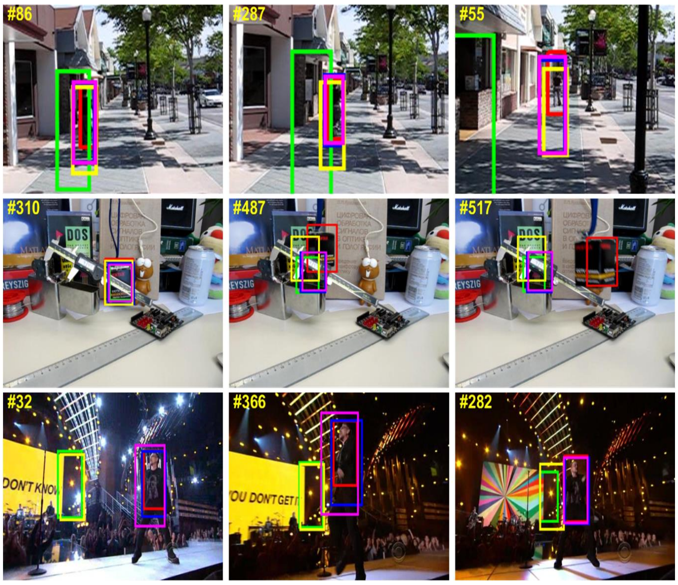


Fig. 1: Tracking in challenging environments including illumination change (Human), heavy occlusion (Box) and complex background (Singer). The results of the cfnet [2], Staple [3], DLSSVM [4], SRDCF [5] tracking methods and our tracker are represented by green, blue, yellow, magenta and red rectangles, respectively.

Recently, convolutional neural networks (CNNs) based on features have been demonstrated state-of-the-art performance on a wide range of visual recognition tasks [6]–[8]. Due to the robustness of deep features, we employ deep features for representation in our first-stage tracking. And intensity values are used as features for our second-stage tracking because of its simplicity and efficiency. Furthermore, our appearance model exploits the holistic templates (HOG-based detector) to distinguish the target from the background, and the effectiveness of local patches (two-stage tracker) in handling partial occlusion and scale variation.

As for the motion model, the sequential Bayesian estimation based particle filters are employed in our two-stage tracker. In

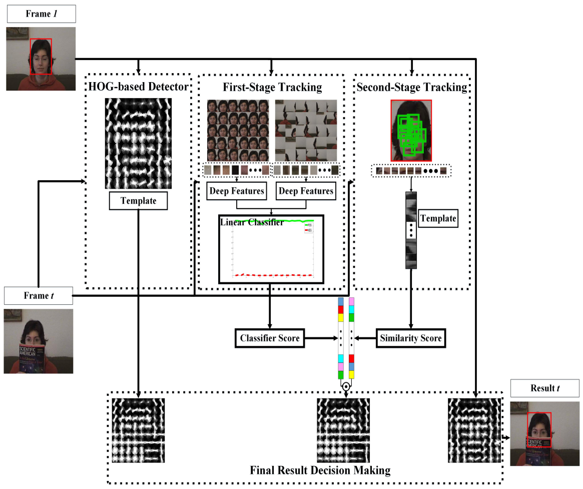


Fig. 2: An overview of the proposed tracking algorithm. The proposed tracker consists of a two-stage tracker and a HOG-based detector. And we ensemble their results by the final result decision making. The figure is best viewed in color.

order to capture appearance variations of the target effectively to alleviate visual drift, we adopt two different update strategies. For the first-stage tracking, the linear classifier is updated per 3 frames to adapt to the rapid variation of the appearance model. For second-stage tracking and HOG-based detector, we set a learning rate to keep a relative static appearance model. After that, the final result decision making is adopted as the post-processor in our tracker. The final result decision making overcomes the shortcoming of stochastic search in the two-stage tracker, and makes up the disadvantage of partial occlusion and scale variation in HOG-based detector.

The major contributions of our framework are as follows:

(1) We propose a robust visual tracking algorithm with a collaborative model which depends on a two-stage tracker and a HOG-based detector;
(2) The final result decision making overcomes the aforementioned disadvantages of the two parts;
(3) Numerous experiments on various challenging sequences show that the proposed algorithm outperforms the state-of-the-art methods.

II. Related Work

Visual tracking aims to identify and localize an unknown target in a video, given the target specified by a bounding box in the first frame. Several datasets and robustness evaluation tools have been displayed in the literature, and in this section we list some recent works related to this work.

A. Appearance Models

For visual tracking, the target object is represented by an appearance model and the tracker estimates the location of the target in each frame. The appearance model can be broadly divided into the generative model and the discriminative model. The generative model based trackers usually search the most similar candidate of the target which corresponds to minimal reconstruction error. Numerous algorithms have been proposed with demonstrated success, e.g., [9], [10]. The discriminative model is to separate the target from background by formulating the tracking problem as a binary classification problem. For the discriminative model, quite a few algorithms have been proposed [4], [11]. And for most discriminative model based online tracker, training samples are selected depending on the location of the target of current frame. In order to maximize the benefits of both, we design a simple yet robust collaborative model which is composed of the generative model and the discriminative model.

B. Correlation Filter based Tracking

Benefiting from the computational efficiency and competitive accuracy, the correlation filter based trackers have drawn more and more attention. The MOSSE [12] runs at 100fps with a minimum sum of the squared error. And Henriques et al. [13] proposed a kernelized correlation filters (KCF) with circulant matrices. To solve the scaling problem in KCF, DSST [14] learns separate filters for translation and scaling. And MUSTer [15] utilizes short-term and long-term strategy to improve the model stability. However, such correlation filter based trackers often suffer from boundary effects. SRDCF [5] utilize a spatial regularization term to solve this problem. Another problem among these trackers is the low discriminative features. C-COT [16], ECO [17] and DeepSRDCF [5] employ hierarchical deep features, which disables the real-time property of the tracker gradually.

C. CNN-based Tracking

Convolutional Neural Network (CNN) has recently drawn a lot of attention in computer vision field due to its strong representation power. Bohyung Han [18] takes full advantage of the end-to-end learning, and has achieved the state-of-the-art performance by updating the network online. In order to improve ability of the inter-class classification, Heng Fan [19] utilizes recurrent neural network (RNN) to model the structure of the object, and combines it into CNN to improve its robustness in face of similar distractors. In addition, Bohyung Han [20] proposes an ensemble based tracker which aims to learn more robust appearance model of the target. Nevertheless, such trackers merely consider the discriminative model, and the generative model has been ignored.

III. The Proposed Collaborative Model

In this section, we first present the two-stage tracker proposed in this work. Then we provide the technical details of particle filters. At last, the HOG-based detector and the final result decision making will be explained. Fig. 2 is an overview of the proposed approach.

A. Two-Stage Tracker

The tracking result is obtained in two stages. In the first stage, the initial tracking result is estimated by an adaptive appearance model. And then in the second stage, the final

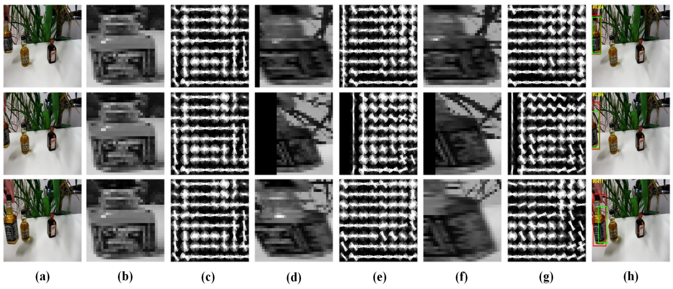


Fig. 3: (a) The 384th, 396th, and 411th frame of Liquor sequence. (b) The template we sampled at first frame. (c) The corresponding HOG feature of the template. (d) The detection results of (a). (e) The corresponding HOG feature of (d). (f) The tracking results of (a). (g) The corresponding HOG feature of (f). (h) The final results of (a). Red boxes show final results and the green ones are results of two-stage tracker.

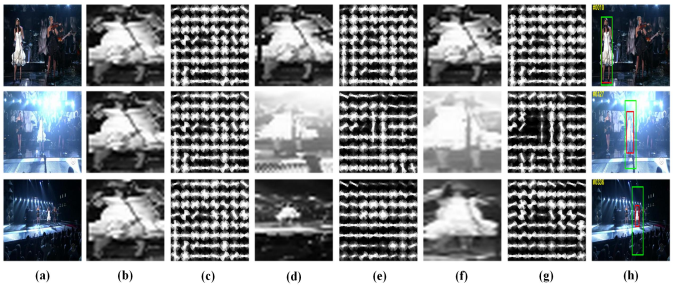


Fig. 4: (a) The 10th, 127th, and 336th frame of Singer1 sequence. (b) The template we sampled at first frame. (c) The corresponding HOG feature of the template. (d) The detection results of (a). (e) The corresponding HOG feature of (d). (f) The tracking results of (a). (g) The corresponding HOG feature of (f). (h) The final results of (a). Red boxes show final results and the green ones are detection results.

tracking result is determined with a relative static appearance model. Particle filters are used in both two stages.

1) First-Stage Tracking: We formulate the first-stage tracking (FT) as a classification problem. A linear classifier is designed to separate the target object from the background. Spatial fine-grained details and semantic information are necessary to achieve this aim. According to [21], features on the ***Conv5-4*** layer of VGG-Net [22] are effective in discriminating the targets even with dramatic background changes thanks to the semantic information, and the ***Conv3-4*** layer of VGG-Net encodes more fine-grained details which is useful to locate target precisely. Therefore, we fuse features of ***Conv5-4*** (upsampling with bilinear interpolation) and ***Conv3-4*** by:

$$\mathscr{F} = \sum_{i} \alpha_i f_i \tag{1}$$

where α_i is the fusion weight, f_i is the feature of ***Conv5-4*** and ***Conv3-4***.

To initialize the classifier in the first frame, we draw positive and negative samples around the location of the target marked with the bounding box. We set the scales of the positive and negative candidates as the same as our labeled target object. To construct the training data, we crop the patches from positive samples and negative samples and extract deep features of each image patch. The loss function of our linear classifier is shown below:

$$\begin{gathered} \mathscr{L}(w) = \frac{1}{M}\sum_{i}^{M}\ell(l_i, w, \mathscr{F}_i) + \frac{\lambda}{2}\|w\|_2^2 \\ \{\mathscr{F}_i\}_{i=1}^{M}, \mathscr{F}_i \in \mathbb{R}^{n+2d}, l_i \in \{+1, -1\} \end{gathered} \tag{2}$$

where $\{\mathscr{F}_i, l_i\}_{i=1}^{M}$ is the training sample, and M is the number of training sample, and w is the classifier parameter, and $\ell(\cdot)$ is a logistic regression loss function, and λ is the regularization term, d is the dimensionality of the image vectors reshaped from the image matrix and n is the number of image vectors. The corresponding classification score is:

$$h(\mathscr{F}) = \frac{1}{1 + e^{-w^{w^\top \mathscr{F}'}}} \tag{3}$$

where $\mathscr{F}^{'} = \left[\mathscr{F}^\top, 1\right]^\top$ is the extended vector of $\mathscr{F}^\top$. A sample with a larger classification score means that it is more likely to be generated from the target class. And we retrain the linear classifier every 3 frames to balance the computational burden and robustness of the adaptive appearance model.

2) Second-Stage Tracking: To reduce the risk of visual drift, we construct a relative static appearance model in second-stage tracking (ST) to refine the FT result.

We warp each input image I to a fixed size. Then we densely sample a set of overlapping local image patches by a sliding window. After the aforementioned pre-processing, we randomly select a number of patches. Given a local patch, the response to input image I is defined as:

$$S_i = P_i \otimes I \tag{4}$$

where P_i is the i-th patch, S_i is the response of P_i. These patches are also used in subsequent frames as templates, so these responses are repeatedly used. In order to improve the representation ability of these templates, we stack all responses S_i in turn to construct a complex template T.

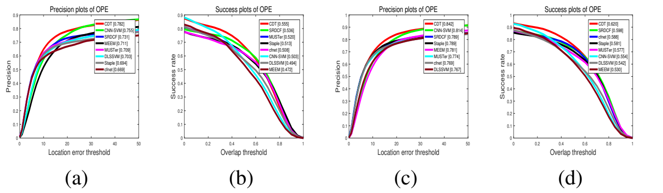
The update strategy of the complex template T is computed by:

$$T_t = (1-\lambda)T_{t-1} + \lambda \hat{T}_{t-1} \tag{5}$$

where T_t is the target template at frame t, $\hat{T}_{t-1}$ is the complex template at frame $t-1$, and λ is the update rate. For the relative static appearance model, we set the update rate λ to 0.98.

After we get the second-stage tracking result, the overall tracking result is computed by:

$$\begin{gathered} Sim = \max_{1 \le i \le n} (h_i(\mathscr{F})^{'} \odot h_i(T)) \\ h_i(T) = e^{-|vec(T_t) - vec(T_t^i)|_2^1} \end{gathered} \tag{6}$$



(a) (b) (c) (d)

Fig. 5: (a) and (b) are the precision and success plots on OTB50, respectively. (c) and (d) are the precision and success plots on OTB100, respectively.

where T_t is the target template at frame t, T_t^i is the i-th candidate sample at frame t, $h_i(\mathscr{F})$ is the i-th classification score of our linear classifier, $h_i(T)$ is the i-th similarity score of our second-stage tracking, $i \in [1, n]$ is the number of particle samples at frame t, and the operator $\odot$ is the Hadamard (element-wise) product.

B. Tracking by Particle Filters

Our two-stage tracker is implemented with particle filters and details are as follows. Given the observation set $O_t = \{o_1, ..., o_t\}$ where t is the index of current frame, our goal is to determine a posteriori probability $p(\mathbf{s}_t|O_t)$ by using the theorem of Bayes:

$$p(\mathbf{s}_t|O_t) = p(\mathbf{o}_t|\mathbf{s}_t)\int p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{s}_{t-1}|O_{t-1})d\mathbf{s}_{t-1} \qquad (7)$$

where $\mathbf{s}_t = [x_t, y_t, s_t]^\top$ is the target state with translations x_t, y_t and scale s_t, and $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ is the motion model that predicts the state $\mathbf{s}_t$ based on the previous state $\mathbf{s}_{t-1}$, and $p(\mathbf{o}_t|\mathbf{s}_t)$ is the appearance model that estimates the likelihood of observation $\mathbf{o}_t$ at the state $\mathbf{s}_t$ belonging to the target category. The motion model between two consecutive frames is assumed to be a Gaussian distribution:

$$p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathscr{N}(\mathbf{s}_t; \mathbf{s}_{t-1}, \sum) \qquad (8)$$

where $\sum$ denotes a diagonal covariance matrix with diagonal elements: σ_x^2, σ_y^2, and σ_s^2. The likelihood $p(\mathbf{o}_t|\mathbf{s}_t)$ is defined as: $p(\mathbf{o}_t|\mathbf{s}_t) \propto Sim^*$ where Sim^* is defined in Eq. 6. Thus, the optimal object state $\mathbf{s}_t^*$ at time t can be determined by solving the following maximum a posterior (MAP) problem:

$$\mathbf{s}_t^* = \underset{\mathbf{s}_t}{\text{argmax}}\, p(\mathbf{s}_t|O_t) \qquad (9)$$

C. HOG-based Detector

The particle filters are based on the stochastic search, which is powerless to re-detect the target when visual drift occurs. To make up the shortcoming, a HOG-based detector is employed in this part [23]. The HOG-based detector is implemented by an exemplar-based linear discriminant analysis classifier:

$$\begin{aligned} H_t(X) &= \text{sign}(\omega_t^\top \times X) \\ \omega_t &= C_t^{-1}(X_t - \mu_t) \end{aligned} \qquad (10)$$

TABLE I: Score of success plots. (The red fonts indicate the best performance, the blue fonts indicate the second best ones and the green fonts marks the third best ones.)

Name Att	CDT	cfnet	Staple	DLSSVM	SRDCF
FM	0.542	0.550	0.507	0.534	0.537
BC	0.643	0.565	0.573	0.531	0.583
MB	0.575	0.553	0.557	0.541	0.608
DEF	0.660	0.544	0.594	0.601	0.625
IV	0.619	0.549	0.598	0.521	0.613
IPR	0.556	0.545	0.534	0.509	0.524
LR	0.552	0.552	0.411	0.386	0.480
OCC	0.579	0.532	0.553	0.511	0.556
OPR	0.589	0.548	0.543	0.523	0.574
OV	0.592	0.423	0.488	0.499	0.460
SV	0.570	0.563	0.522	0.460	0.535
OverAll	0.620	0.588	0.581	0.542	0.598

where X is a positive sample at frame t, and C and μ are the covariance matrix and the mean vector of negative dataset respectively, and the column vector ω represents the object, and (C, μ) represents the background.

Each exemplar is described by the HOG feature, considering its successful applications in object detection. The final HOG-based detector is computed by:

$$\begin{aligned} H(X) &= H_1(X) + \sum_i \alpha_i H_i(X) \\ \alpha_i &= \frac{H_i(X)}{H(X)} \end{aligned} \qquad (11)$$

where $H_i(X)$ is the result of frame i, α_i is a update rate.

D. Final Result Decision Making

Our two-stage tracker just employs a simple dynamic model based on the stochastic search. And the tracking part of our algorithm cannot maintain useful target information during the entire tracking sequences, which fails to re-detect the target when the target reappears after visual drift. In this paper, we ensemble the result of the detector and the result of the tracker in final result decision making.

The first frame is always important in tracking, because it includes the only precise label in the tracking process. Furthermore, our two-stage tracker merely considers the local information of the target, and the global information of the target has been ignored. That is, the processing of local patches makes the two-stage tracker lack of global information in edges and structures, whether in the first stage or in the second stage. On the contrary, the HOG descriptor in our detector focuses on edges and structures, which are less sensitive to the change of view and rotation of the target. With the global information of HOG-based detector, our tracker contains both local and global information of the target.

In the process of implementation, we resample the tracking result map to a fixed size (64×64) with bilinear interpolation. Then we extract the HOG feature from resized map and the first frame template, marked with $H(Sim)$ and $H(F)$, respectively. After that, we measure the similarity by the squared distance:

$$
\begin{aligned}
D_1 &= \|vec(H(Sim)) - vec(H(F))\|_2^2 \\
D_2 &= \|vec(H(X)) - vec(H(F))\|_2^2
\end{aligned}
\tag{12}
$$

where $vec(\cdot)$ is a column vector by concatenating all the elements in $(\cdot)$. If $D_1 < D_2$, we select the result of the detector as our final result for current frame. Otherwise, the result of the tracker is the final result. We show an example that the target occurs out of view and reappears again in Fig. 3.

As we can see in Fig. 3, the final result that benefits from decision making performance is better than the result of two-stage tracker. When the target disappears from the scenario, our two-stage tracker merely employs stochastic search and a simple model update strategy, which results in the tracking drift phenomenon. And training samples in limited quantities in FT can easily cause tracking failure in such occasion.

Although enlarging the search region of the sampling of particles and adding more candidate particles increases the accuracy of the two-stage tracker, it adds computational cost of our two-stage tracker. Therefore, we set the detection region bigger than tracking to provide global information and additional search samples. As a result, the particles can still move towards the target object by using the supervision signal of HOG-detector.

We show another example that the target has the scale variation. As is shown in Fig. 4, our final tracker performs better than HOG-based detector when the scale variation occurs. It is because that our two-stage tracker employs local features extracted from the normalized local image patches and our relative static appearance model in second-stage tracking plays an important role in the target template to handle the drift problem.

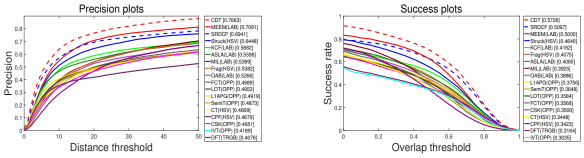
IV. Experiment

To evaluate our approach CDT, we perform comprehensive experiments on three benchmark datasets: OTB-50, OTB-100 [24] and Temple-Color [25]. The OTB-50 and OTB-100 include 50 and 100 sequences respectively with 11 attributes. The Temple-Color dataset contains 128 RGB sequences with annotations of challenging factors.

A. Experimental Setting

The proposed RLSVM is implemented in MATLAB with Matconvnet toolbox [26].

For the detector, we normalize the size of detecting window to 64 × 64 pixels and then extract the HOG features with a cell size of 8 pixels and 9 orientation bins. And the detection area is set to 25. For the two-stage tracker, the number of particles is set to 600 in all experiments. Each image is normalized to 32 × 32 pixels. In FT, we extract overlapping 16 × 16 patches with a shift of 8 pixels. The deep features are extracted from VGG-Net-19, which is trained on ImageNet [27]. And we set value of α to 0.5 and 1 for the ***Conv5-4***, ***Conv3-4*** layers, respectively. In ST, we set the sliding window to 6 × 6 pixels. The number of patches randomly sampled is 150.



(a) (b)

Fig. 6: The Error scores of the precision plots and the overlap scores of the success plots in Template-Color dataset.

B. Experiments on OTB-50 and OTB-100 Dataset

We perform a comprehensive comparison with 7 recent state-of-the-art trackers: SRDCF [5], MUSTer [15], Staple [3], cfnet [2], CNN-SVM [28], DLSSVM [4] and MEEM [29].

A comparison with the trackers mentioned above on the OTB-50 and OTB-100 is shown in Fig. 5. We report the score of the success plot and the score of the precision plot on both datasets. Our approach achieves the results with the precision score of 0.782 and 0.842 on OTB-50 and OTB-100, and the success score of 0.555 and 0.620 on OTB-50 and OTB-100, which is the best among the 8 trackers.

C. Attribute based analysis

In the OTB-100 dataset, different videos are annotated with 11 different attributes. TABLE I contains success scores of the 11 different attributes on the OTB-100 dataset. As TABLE I shows, our CDT tracker ranks high on 11 attributes in success plots. The reason why our CDT tracker achieves outstanding performance are analyzed as follows. (1) In challenging scenarios of scale variations, in-plane rotations and out-of-plane rotations, inaccurate estimation of the target often leads to the inclusion of misaligned training samples. Our collaborative model is capable of alleviating the impact of such samples, thereby lowering the risk of drift and tracking failure. (2) Our method is effective in handling background clutters because of features with semantic information and spatial details from the hierarchical layers of CNNs in our second-stage tracker. (3) Our CDT tracker takes advantage of normalized image information from HOG-based detector and two-stage tracker, which is robust to illumination variation.

D. Experiments on Temple-Color Dataset

Finally, we perform experiments on the Temple-Color dataset with 128 videos. A comparison with state-of-the-art trackers in precision plots is shown in Fig. 6(a). Among the compared methods, our approach improves the state-of-the-art on this dataset with an error score of 0.7663. Fig. 6(b) shows the success plot over all the 128 videos in the Temple-Color dataset. Our tracker outperforms state-of the-art approaches with an AUC score of 0.5726.

E. Failure Cases

We show a few failure cases in Fig. 7 For the *Biker*, the *GragonBaby* and *Skater2* sequences, when the appearance of the target changes quickly, the proposed tracker fails to

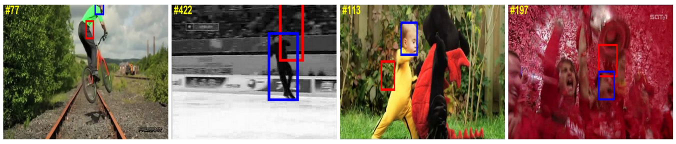


Fig. 7: Failure cases (*Biker*, *Skater2*, *GragonBaby* and *Soccer*). Red boxes show our results and the blue ones are ground truth.

follow targets as the simple update strategy of the proposed method updates not that fast. For the *Soccer* sequence, our tracker is not effective to distinguish the foreground from the background. The reason is that the HOG descriptor in our final result decision making fails to deal with noisy edge regions.

V. Conclusion

In this paper, we propose an online tracking algorithm based on collaboration model (CDT). In our tracker, the holistic template constructed by HOG-based detector can effectively deal with out-of-view and fast motion cases. In our two-stage tracker, FT extracts the deep features of local patches and learns a linear classifier. The result of FT is refined by ST to reduce the risk of visual drift. The two-stage tracker achieves better performance in cases of occlusion and scale variation. And the final result decision making enables the holistic template and the local model to make up the shortage of each other. Experiments on several challenging sequences with comparisons to state-of-the-art tracking methods demonstrate the effectiveness of our tracking algorithm. Moreover, our two-stage tracker is a very flexible framework and our implementation is far from optimal. And we will investigate to extend this work to design more efficient tracking algorithms in the future.

Acknowledgment

This work was supported in part by the Natural Science Foundation of China under Grant 61472102, and 61672194, by the Distinguished Youth Science Foundation of Heilongjiang Province of China under Grant JC2018021, by the Shandong Provincial Natural Science Foundation, China under Grant ZR2016FM04, and by the Humanity and Social Science Youth foundation of Ministry of Education of China under Grant 14YJC760001.

References

[1] N. Wang, J. Shi, D. Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *IEEE International Conference on Computer Vision*, 2015, pp. 3101–3109.

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

[3] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. Torr, "Staple: Complementary learners for real-time tracking," vol. 38, no. 2, pp. 1401–1409, 2015.

[4] J. Ning, J. Yang, S. Jiang, L. Zhang, and M. H. Yang, "Object tracking via dual linear structured svm and explicit feature map," in *Computer Vision and Pattern Recognition*, 2016, pp. 4266–4274.

[5] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318.

[6] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[7] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask r-cnn," 2017.

[8] Z. Shen, Z. Liu, J. Li, Y. G. Jiang, Y. Chen, and X. Xue, "Dsod: Learning deeply supervised object detectors from scratch," in *IEEE International Conference on Computer Vision*, 2017, pp. 1937–1945.

[9] X. Mei and H. Ling, "Robust visual tracking using l(1) minimization," in *IEEE International Conference on Computer Vision*, 2009, pp. 1436–1443.

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

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

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

[13] J. F. Henriques, C. Rui, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[14] M. Danelljan, G. Hger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference*, 2014, pp. 65.1–65.11.

[15] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Computer Vision and Pattern Recognition*, 2015, pp. 749–758.

[16] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*, 2016, pp. 472–488.

[17] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," pp. 6931–6939, 2017.

[18] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Computer Vision and Pattern Recognition*, 2016, pp. 4293–4302.

[19] H. Fan and H. Ling, "Sanet: Structure-aware network for visual tracking," pp. 2217–2224, 2016.

[20] B. Han, J. Sim, and H. Adam, "Branchout: Regularization for online ensemble tracking with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 521–530.

[21] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *IEEE International Conference on Computer Vision*, 2016, pp. 3074–3082.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, 2014.

[23] C. Gao, F. Chen, J. G. Yu, R. Huang, and N. Sang, "Robust visual tracking using exemplar-based detectors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 2, pp. 300–312, 2017.

[24] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[25] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans Image Process*, vol. 24, no. 12, pp. 5630–5644, 2015.

[26] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," pp. 689–692, 2014.

[27] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248–255.

[28] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," pp. 597–606, 2015.

[29] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," vol. 8694, pp. 188–203, 2014.