High-Performance Discriminative Tracking with Target-Aware Feature Embeddings

Bin Yu[1,2](✉), Ming Tang[2], Linyu Zheng[1,2], Guibo Zhu[1,2], Jinqiao Wang[1,2,3], and Hanqing Lu[1,2]

[1] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[2] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No.95, Zhongguancun East Road, Beijing 100190, China
{bin.yu,tangm,linyu.zheng,gbzhu,jqwang,luhq}@nlpr.ia.ac.cn
[3] ObjectEye Inc., Beijing, China

Abstract. Discriminative model-based trackers have made remarkable progress recently. However, due to the extreme imbalance of foreground and background samples, the learned model is hard to fit the training samples well in the online tracking. In this paper, to alleviate the negative influence caused by the imbalance issue, we propose a novel construction scheme of target-aware features for online discriminative tracking. Specifically, we design a sub-network to generate target-aware feature embeddings of foregrounds and backgrounds by projecting the learned feature embeddings into the target-aware feature space. Then, a model solver, which is integrated into our networks, is applied to learn the discriminative model. Based on such feature construction, the learned model is able to fit training samples well in the online tracking. Experimental results on four benchmarks, OTB-2015, VOT-2018, NfS, and GOT-10k, show that the proposed target-aware feature construction is effective for visual tracking, leading to the high-performance of our tracker.

Keywords: Visual tracking · Data imbalance · Discriminative model

1 Introduction

Visual tracking is one of the fundamental problems in computer vision, spanning a wide range of applications including video understanding, surveillance, and robotics. Despite significant progress in recent years [2,5,17,30,35], visual

B. Yu—The first author is a student.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-88004-0_1) contains supplementary material, which is available to authorized users.

H. Ma et al. (Eds.): PRCV 2021, LNCS 13019, pp. 3–15, 2021.
https://doi.org/10.1007/978-3-030-88004-0_1

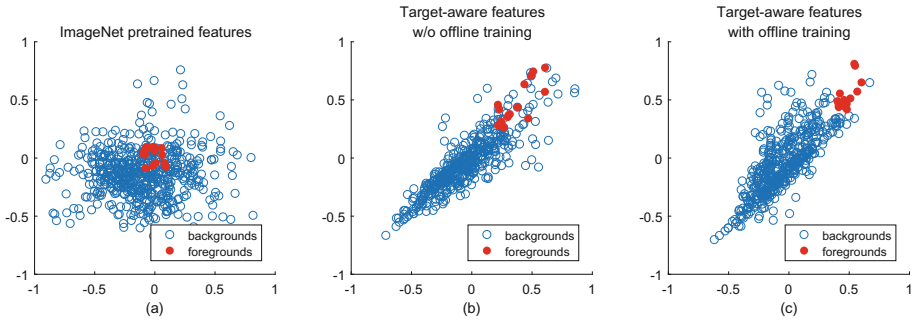

Fig. 1. Distributions of the foreground and background samples with three different kinds of features. It can be seen that from (a) to (c), it is from hard to easy for a discriminative model to fit training samples well in online tracking.

tracking is still a challenge due to several severe interferences (*i.e.*, large appearance changes, occlusions, and background clutters) and very limited training samples.

Different from Siamese based trackers [17,30,32], online discriminative trackers [2,6,23,27–29,33–35] can effectively exploit the background information online, and thus have achieved the state-of-the-art results on multiple challenging benchmarks [11,16,31]. However, in the online tracking, the extreme imbalance of foreground and background samples causes the learned discriminative model to pay more attention to backgrounds while less to foregrounds, and further to be difficult to fit training samples well (see Fig. 1(a)). These problems negatively affect the discriminative power of the trackers. In fact, there are already methods before attempting to solve the problems by mitigating the emphasis on background information in the learned models, such as sample re-weighting using Gaussian-like maps [7], spatial reliability maps [22] or binary maps [14]. However, they inevitably increase the complexity of model learning, and thus affect the tracking efficiency. Besides, they are hard to take advantage of the end-to-end CNNs training like [2,5,29,35] to improve their accuracy and robustness further.

Different from the previous methods, in this paper, we attempt to alleviate the ahead-mentioned problems from the perspective of feature construction. We consider the feature embeddings used in most previous trackers are target-unaware ones where the construction of feature is independent of whether a sample is from the foreground or not. With such features, the distributions of foreground and background samples which are all extracted over the image region overlap severely (as shown in Fig. 1(a)), making it hard for the discriminative model to fit foreground and background samples well. Note that although there exists methods [4,19] that aim at constructing target-aware features for visual tracking, both are designed for Siamese-based tracking schemes and may not be suitable for online discriminative ones.

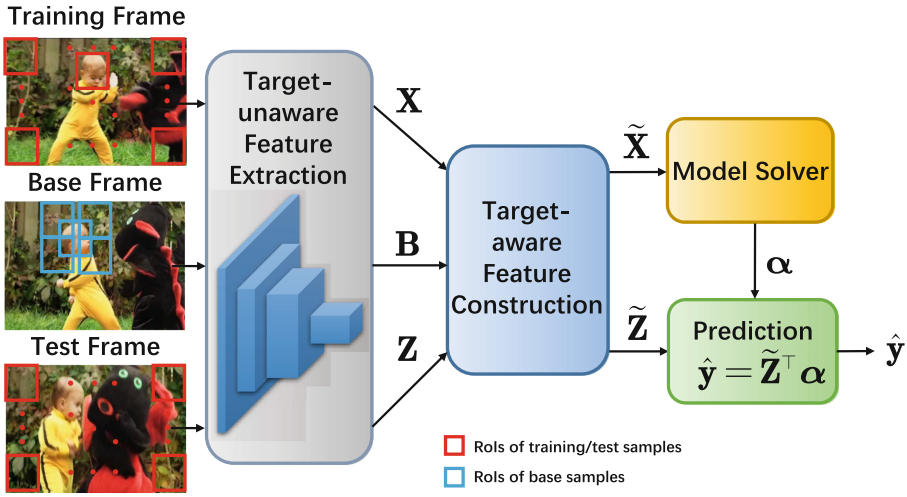

Fig. 2. An illustration of the proposed discriminative tracking scheme with target-aware feature construction.

Based on the above observation, we propose a simple yet powerful scheme of target-aware feature construction for online discriminative tracking. First, we introduce a set of *base samples*, $\{\mathbf{b}_i\}$, which are extracted around the target over a smaller region than that where training samples are extracted (see Fig. 2), to obtain target-specific information directly. Then, target-aware feature embeddings are constructed by mapping the target-unaware ones of training samples to the target-aware space with the dot products $\langle \cdot, \mathbf{b}_i \rangle$. In this way, the generated features are composed of similarities to base samples, causing the overlap between the distributions of foreground samples and those of background samples decreases in the feature space (see Fig. 1(b)). Based on such feature construction, the learned discriminative model is able to distinguish between foreground and background samples better.

However, off-the-shelf CNNs features of training samples and base ones may not be suitable enough for our target-aware feature construction since the extractions of those features are independent of our scheme. To this end, we design a sub-network to construct the target-aware features, enabling an end-to-end manner to learn target-unaware feature embeddings. Concretely, as shown in Fig. 2, first we extract the target-unaware features of training samples and base ones from the input frames, respectively, with separate branches in the proposed networks. Then, the target-aware feature embeddings are obtained through a target-aware feature construction sub-network, and a differentiable model solver [1,35], which is integrated into our networks, is applied to learn the discriminative model. Given a test frame, the target-aware feature embeddings of test samples are obtained in the same way and the labels of these samples are predicted with the learned discriminative model. In this way, our networks can be trained in an end-to-end manner, enabling learning the feature embeddings that are

tightly coupled with our novel target-aware feature construction and the discriminative tracking scheme. With such target-aware feature embeddings, the overlap between the distributions of foregrounds and that of backgrounds further decreases in the feature space (see Fig. 1(c)), leading to strong discrimination of the learned model.

Finally, based on the learned target-unaware feature extraction network and the target-aware feature construction, an effective discriminative tracker, TADFT, is developed and evaluated on four public benchmarks, OTB-2015 [31], VOT-2018 [16], NfS [13], and GOT-10k [11], achieving state-of-the-art results.

In summary, our main contributions are in three folds.

1. We present a novel construction scheme of target-aware features for online discriminative tracking to alleviate the negative influence of imbalance of foreground and background samples.
2. We design a sub-network to construct the target-aware features of foregrounds and backgrounds, benefiting an end-to-end way to learn the target-unaware feature embeddings.
3. We develop a discriminative tracker TAFDT based on our target-aware feature construction and extensively evaluate TAFDT on four public benchmarks.

2 Discriminative Tracking with Target-Aware Feature Embeddings

2.1 Target-Unaware Feature Extraction

To achieve strong discrimination of target-unaware features, each image is processed by a target-unaware feature extraction network consisting of a backbone, *i.e.*, ResNet [9], and a head network [35]. Concretely, for each input training/test frame, N RoIs are first generated by uniform sampling across the whole image region. Then we extract Block-3 and Block-4 feature maps of ResNet and pass them through two convolutional layers to obtain two feature maps. Two 512-dimensional feature vectors of each RoI are finally obtained by using PrPool layers [12] and fully-connected layers. Thus, the feature dimension D of each sample is 1024 (concatenated by the two feature vectors). As illustrated in Fig. 2, we obtain $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ and $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N] \in \mathbb{R}^{D\times N}$ consisting of N D-dimensional training samples and test ones, repectively.

2.2 Target-Aware Feature Construction

In order to obtain target-specific information, we extract base samples around the target object over a smaller region than that where training samples are extracted (see Fig. 5). Given an additional input frame, called base frame, we obtain $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_M] \in \mathbb{R}^{D\times M}$ through the target-unaware feature extraction network, where $\mathbf{b}_i$ is a D-dimensional base sample.

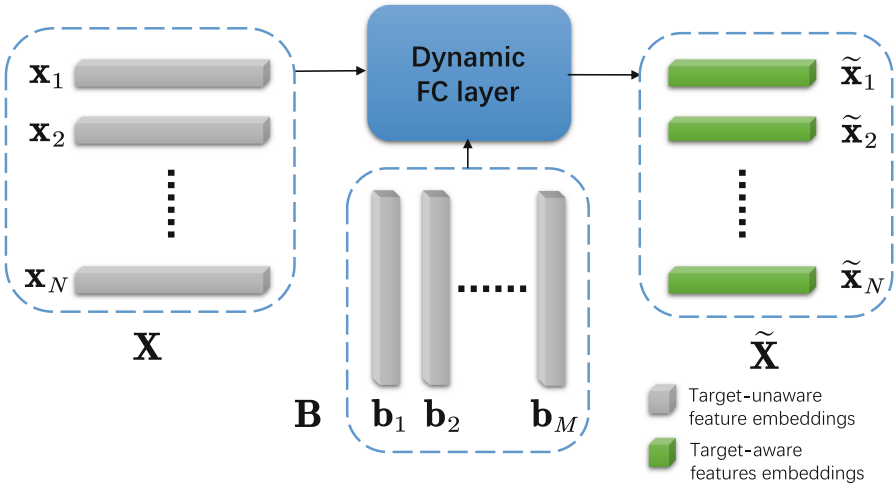

Fig. 3. Construction of target-aware features. It is performed by a dynamic filters layer, using the base sample matrix $\mathbf{B}$ as the weights of a fully-connected layer. For a single training/test sample, the dimensionality of features is transformed from D to M.

Then, given a D-dimensional sample $\mathbf{x}$, the process of the target-aware feature construction is represented as follows,

$$\widetilde{\mathbf{x}} = [\langle \mathbf{x}, \mathbf{b}_1 \rangle, \langle \mathbf{x}, \mathbf{b}_2 \rangle, ..., \langle \mathbf{x}, \mathbf{b}_M \rangle]^\top = \mathbf{B}^\top \mathbf{x}, \tag{1}$$

where $\langle , \rangle$ represents dot product.

Thus, we are able to obtain the target-aware feature embeddings of training samples and test ones by

$$\begin{aligned} \widetilde{\mathbf{X}} &= [\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, ..., \widetilde{\mathbf{x}}_N] = \mathbf{B}^\top \mathbf{X}, \\ \widetilde{\mathbf{Z}} &= [\widetilde{\mathbf{z}}_1, \widetilde{\mathbf{z}}_2, ..., \widetilde{\mathbf{z}}_N] = \mathbf{B}^\top \mathbf{Z}. \end{aligned} \tag{2}$$

Further, to obtain target-aware feature embeddings efficiently and enable learning target-unaware ones in an end-to-end way, we view the projection operation as a dynamic fully connected layer, whose weights are equal to the generated base sample matrix $\mathbf{B}$, as shown in Fig. 3. As such, we are able to construct a specific feature space for each target object by changing the input base frame.

2.3 Ridge Regression with Target-Aware Feature Embeddings

Ridge Regression with Single Frame. As a classical discriminative model, ridge regression model has been confirmed to be simple yet effective in the field of visual tracking by many trackers [6,10,14,35], and thereby employed in our approach.

The optimization problem of ridge regression with target-aware feature embeddings is formulated as follows,

$$\min_{\boldsymbol{\alpha}} \|\widetilde{\mathbf{X}}^{\top}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \lambda||\boldsymbol{\alpha}||_2^2, \tag{3}$$

where $\mathbf{y} \in \mathbb{R}^{N\times 1}$ is the groundtruth labels and $\lambda \geq 0$ is a regularization parameter. The optimal $\boldsymbol{\alpha}^*$ can be analytically solved by

$$\boldsymbol{\alpha}^* = \left(\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^{\top} + \lambda\mathbf{I}\right)^{-1}\widetilde{\mathbf{X}}\mathbf{y}, \tag{4}$$

where $\mathbf{I} \in \mathbb{R}^{D\times D}$ is the identity matrix.

Note that in the optimization problem of ridge regression with the D-dimensional target-unaware feature embeddings , we achieve the optimal solution with Gauss elimination method and its time complexity is $O(D^3/2)$. It means the time-consuming grows cubically with the dimension of the feature embeddings. However, due to the common characteristic in CNNs that deep features are high-dimensional and $M < D$ in our approach ($M = 23 \times 23$ and $D = 31 \times 31$), a faster solution of the optimization problem can be achieved, *i.e.*, $\boldsymbol{\alpha}$ is obtained with time complexity of $O(M^3/2)$ in our approach.

Ridge Regression with Historical Frames. Since the size of discriminative model is reduced by the target-aware feature construction, the influence of the online optimization on efficiency is weaken, and we are able to extend our model with historical frames for robust tracking without increasing heavy computational burden. To be specific, we draw training samples from p historical frames in the training set, *i.e.*, $\{\mathbf{X}_j\}_{j=1}^{p}$. We consider the optimization problem of multiple frames in our approach as follows,

$$\min_{\boldsymbol{\alpha}_p} \frac{1}{p}\sum_{j=1}^{p} \|\widetilde{\mathbf{X}}_j^{\top}\boldsymbol{\alpha}_p - \mathbf{y}\|_2^2 + \lambda||\boldsymbol{\alpha}_p||_2^2, \tag{5}$$

where $\widetilde{\mathbf{X}}_j = \mathbf{B}^{\top}\mathbf{X}_j$.

The objective function in Eq. (5) is minimized by

$$\boldsymbol{\alpha}_p^* = \left(\sum_{j=1}^{p}\widetilde{\mathbf{X}}_j\widetilde{\mathbf{X}}_j^{\top} + \lambda\mathbf{I}\right)^{-1}\sum_{j=1}^{p}\widetilde{\mathbf{X}}_j\mathbf{y}. \tag{6}$$

Similar to DCFST [35], the model solver, *i.e.*, solving for $\boldsymbol{\alpha}_p$ by directly using Eq. (6), can be integrated into the networks as a layer with forward and backward processes during offline training, benefiting an end-to-end manner to learn the target-unaware feature embeddings coupled with our approach.

2.4 Offline Training

To take advantage of multiple frames from the sequence, during offline training, each input image tuple consists of 3 training frames, 3 test ones, and a base one sampled in a sequence. Then the training sample matrices $\{\mathbf{X}_j\}_{j=1}^3$, the test sample matrices $\{\mathbf{Z}_j\}_{j=1}^3$, and the base sample matrix $\mathbf{B}$ are obtained through the target-unaware feature extraction, respectively.

After obtaining $\boldsymbol{\alpha}_p^*$ over the three training frames by Eq. (6), we calculate the predicted labels $\hat{\mathbf{y}}_j \in \mathbb{R}^{N\times 1}$ by $\hat{\mathbf{y}}_j = \widetilde{\mathbf{Z}}_j^\top \boldsymbol{\alpha}_p^*$, where $\widetilde{\mathbf{Z}}_j = \mathbf{B}^\top \mathbf{Z}_j$. In the end, we adopt the modified shrinkage loss like in [21,35] over the three test frames. The objective loss used for offline training is given below.

$$\mathcal{L} = \sum_{j=1}^{q} \frac{\exp(\mathbf{y}) \cdot \|\hat{\mathbf{y}}_j - \mathbf{y}\|^2}{1 + \exp(c_1 \cdot (c_2 - |\hat{\mathbf{y}}_j - \mathbf{y}|))}, \tag{7}$$

where $c_1 = 10$ and $c_2 = 0.2$ are set the same as in [21,35].

It is worth noting that even though the shrinkage loss and our target-aware feature construction both aim at alleviating the problems caused by the imbalance of foreground and background samples, they focus on different aspects. The shrinkage loss focuses on the imbalance issue of samples from test frames in the offline training of the network and is used to prevent the vast number of easy background samples from overwhelming the learning of feature embeddings [35]. Different from that, the target-aware feature construction focuses on the problems caused by the imbalance of samples from training frames in model learning and enables the learned discriminative model fit training samples well in the online tracking.

2.5 Online Tracking

Updating. Several recent tracking approaches [29,34,35] use a moving average strategy to update the discriminative model. Despite its simplicity, this strategy limits the flexibility of the update mechanism due to synthetic appearance and a constant update rate across the whole sequence. To address the drawbacks of this strategy, our approach allows the discriminative model to be easily updated by adding new frames to the training set as in [2]. We ensure a maximum memory size of 50 for the training set. During online tracking, the model is updated every 15 frames. Besides, we update the base sample matrix in every frame as follows.

$$\begin{aligned} \overline{\mathbf{B}}_T &= \mathbf{B}, \quad T = 1, \\ \overline{\mathbf{B}}_T &= (1-\gamma)\overline{\mathbf{B}}_{T-1} + \gamma \mathbf{B}, \quad T > 1, \end{aligned} \tag{8}$$

where γ is a weight parameter and T is the frame number.

Localization. In a new test frame, given the optimal solution to Problem (5), $\boldsymbol{\alpha}_p^*$, the updated base sample matrix $\overline{\mathbf{B}}$ and the test sample matrix $\mathbf{Z}$, we predict the target location by $\hat{\mathbf{y}} = \widetilde{\mathbf{Z}}^\top \boldsymbol{\alpha}_p^*$, where $\widetilde{\mathbf{Z}} = \overline{\mathbf{B}}^\top \mathbf{Z}$. The sample corresponding to the maximum element of $\hat{\mathbf{y}}$ is regarded as the target object.

Refinement. After locating the target in a new test frame, we refine its bounding box by ATOM [5] for more accurate tracking.

3 Experiments

Our tracker is implemented in Python using PyTorch. On a single TITAN RTX GPU, employing ResNet-50 [9] as the backbone network, TAFDT achieves a tracking speed of 30 FPS. We validate it by performing comprehensive experiments on four benchmarks: OTB-2015 [31], VOT-2018 [16], NfS [13], and GOT-10k [11].

Table 1. Comparisons of different feature constructions for TAFDT on the combined NfS and OTB-2015 benchmarks.

	Ours	Larger area		Smaller area		Baseline
Method	Target-aware	None-aware	Target-aware+	Target-aware−	Center-aware	Target-unaware
Sample area	3×3	5×5	4×4	2×2	1.5×1.5	–
Mean DP(%)	**81.2**	78.8	80.6	79.7	80.4	79.0
AUC (%)	**68.5**	64.9	66.2	66.1	64.0	64.7

3.1 Implementation Details

Training Data. We use the training splits of large-scale tracking datasets including TrackingNet [24], GOT-10k [11], and LaSOT [8]. The networks are trained with 20 image tuples sampled from a random segment of 50 frames in a sequence. For each training/test frame, the search area is 5×5 times the target area. Base samples are extracted from the region centered at the target, with an area of 3×3 times the target area. In addition, we use image flipping and color jittering for data augmentation.

Network Learning. We freeze the weights of the ResNet-50 [9] backbone network during training. The networks are trained for 50 epochs with 1500 iterations per epoch within 30 h. We use ADAM [15] with learning rate decay of 0.2 every 15 epochs, using a initial learning rate of 0.005.

3.2 Feature Comparisons

In order to confirm the effectiveness of the proposed target-aware feature construction, we compare it with other four feature constructions on the combined OTB-2015 and NfS benchmarks. To compare equitably, the only difference between these feature constructions is the area of the region where base samples are extracted. As shown in Table 1, applying none-aware features only obtains similar results to those of the baseline applying target-unaware features directly from the feature extraction network. Applying target-aware+ features obtains

better results due to the reduction of redundant background information, with the mean AUC score of 66.2% and DP of 80.6%. Our tracker with target-aware features further improves the results, giving absolute gains of 2.3% and 0.6% in mean AUC scores and DP, respectively. Note that when more background information is contained in base samples as in target-aware+ features and none-aware ones, it is hard for the discriminative model to pay attention to foreground and fit training samples well. In contrast, when less foreground information is contained in base samples as in center-aware features and target-aware− ones, the results degenerate because the discriminative model is less robust to variation of target appearance. The results show that the area of 3×3 times the target size is the most suitable for our target-aware feature construction, which is the maximum area where each base sample contains part of the target object.

3.3 State-of-the-Art Comparisons

OTB-2015. Figure 4 and Table 2 show the comparisons of TAFDT with the state-of-the-art online discriminative trackers. TAFDT achieves the AUC score

Table 2. The mean overlap of TAFDT and other nine state-of-the-art online discriminative trackers on OTB-2015.

Tracker	TAFDT	ECO [6]	ATOM [5]	DiMP50 [2]	DCFST50 [35]
Venue	Ours	CVPR2017	CVPR2019	ICCV2019	ECCV2020
Mean OP(%)	88.9	84.2	83.4	84.6	87.3
FPS	30	6	30	30	25
Tracker	DCFST18 [35]	VITAL [26]	fdKCF [34]	MDNet [25]	DeepSTRCF [18]
Venue	ECCV2020	CVPR2018	ICCV2019	CVPR2016	CVPR2018
Mean OP(%)	87.2	85.7	82.7	84.7	84.5
FPS	40	2	24	1	5

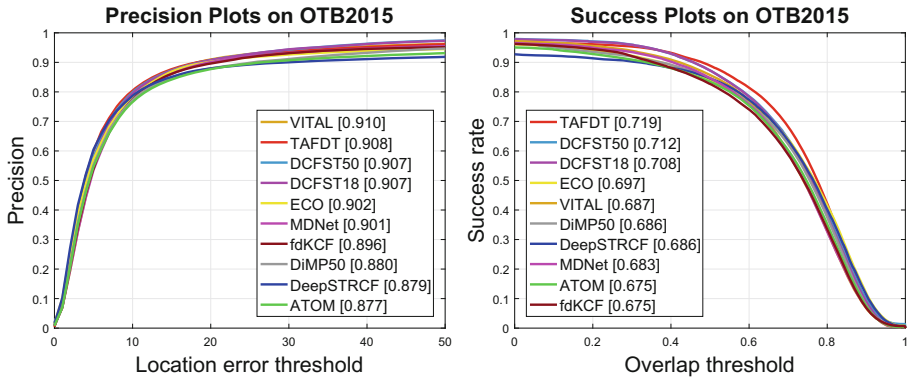

Fig. 4. The mean precision and success plots of the proposed TAFDT and other six state-of-the-art online discriminative trackers on OTB-2015. The mean distance precisions and AUC scores are reported in the legends.

of 71.9%, outperforming the latest DCFST50 [35] and DiMP50 [2] by 0.7% and 3.3%, respectively. In addition, TAFDT achieves the mean OP of 88.9% (see Table 2), leading the second best tracker DCFST50 with an absolute gain of 1.6%. The results show that the target-aware feature construction helps our new discriminative tracker TAFDT achieve new state of the art on this benchmark, while running at 30 FPS, comparable with other trackers.

Table 3 shows the comparison of TAFDT with the Siamese network based trackers including the latest target-aware feature-based trackers MLT [4] and TADT [19]. TAFDT achieves a gain of 1.8% in AUC scores compared with the state-of-the-art SiamRCNN [30] which employs stronger backbone network (ResNet-101) and more training datas (*e.g.*, COCO [20]), respectively. Moreover, TAFDT surpasses the previous 'target-aware' trackers TADT and MLT with significant gains of 5.9% and 10.8%, respectively, confirming the effectiveness and superiority of our target-aware feature construction.

Table 3. The mean AUC scores of TAFDT and other four state-of-the-art Siamese-based trackers on OTB-2015.

Tracker	TAFDT	SiamRCNN [30]	SiamRPN++ [17]	MLT [4]	TADT [19]
Venue	Ours	CVPR2020	CVPR2019	ICCV2019	CVPR2019
AUC(%)	71.9	70.1	69.6	61.1	66.0

Table 4. State-of-the-art comparisons on the VOT-2018 in terms of expected average overlap (EAO), accuracy and robustness.

Tracker	UPDT [3]	SiameseRPN++ [17]	ATOM [5]	DiMP50 [2]	DCFST50 [35]	SiamRCNN [30]	TAFDT
Venue	ECCV2018	CVPR2019	CVPR2019	ICCV2019	ECCV2020	CVPR2020	Ours
EAO	0.378	0.414	0.401	0.440	0.452	0.408	0.455
Robustness	0.184	0.234	0.204	0.153	0.136	0.220	0.180
Accuracy	0.536	0.600	0.590	0.597	0.607	0.609	0.611

VOT-2018. We evaluate TAFDT on VOT-2018, then compare its accuracy, robustness, and EAO score with six state-of-the-art trackers. As shown in Table 4, the proposed TAFDT achieves 45.5% and 61.1% in terms of EAO and accuracy, respectively, surpassing the second best DCFST50 by 0.3% and 0.4% in terms of EAO and accuracy, respectively.

NfS. We evaluate the proposed TAFDT on the 30 FPS version of NfS. Table 5 shows the comparison with six state-of-the-art trackers. Our approach achieves the AUC score of 65.5%. TAFDT consistently outperforms DCFST50 and SiamRCNN with gains of 1.4% and 1.6% in AUC scores, respectively.

GOT-10k. We evaluate the proposed TAFDT on the test set of GOT10k including 180 test videos. In contrast to other datasets, trackers are restricted to use only the training split of the dataset to ensure a fair comparison. Accordingly, we retrain our networks by using only the training split. Figure 5 shows the success plots of TAFDT and other nine state-of-the-art trackers. The success rates at overlap threshold 0.5 are shown in legend which represent the percentages of successfully tracked frames. TAFDT outperforms all previous trackers, achieving an absolute gain of 0.5% in success rates over the previous best method, DCFST50.

Table 5. State-of-the-art comparison on the 30 FPS version of NfS in terms of AUC scores.

Tracker	TAFDT	ECO [6]	UPDT [3]	ATOM [5]	DiMP50 [2]	DCFST50 [35]	SiamRCNN [30]
Venue	Ours	CVPR2017	ECCV2018	CVPR2019	ICCV2019	ECCV2020	CVPR2020
AUC(%)	65.5	46.6	53.7	58.4	62.0	64.1	63.9

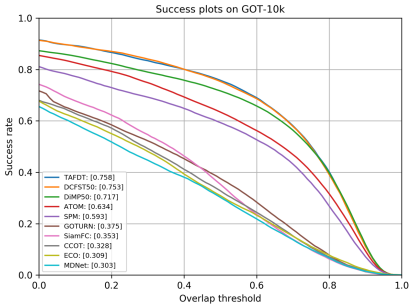

Fig. 5. Success plots of the proposed TAFDT and other nine state-of-the-art trackers on GOT-10k. The success rates at overlap threshold 0.5 are shown in legend.

4 Conclusion

In this paper, we propose a simple yet powerful construction scheme of target-aware features for online discriminative tracking and a target-aware features-based online discriminative tracker, TAFDT. Based on our target-aware feature embeddings, the learned discriminative model is capable of being less affected by the imbalance issue, and thus fitting training samples well in online tracking. The devised networks are able to learn the target-unaware feature embeddings that are tightly coupled with our novel target-aware feature construction and the discriminative tracking scheme. We extensively validate the proposed TAFDT on four public benchmarks. Experimental results verify that target-aware feature construction is effective and leads TAFDT to achieve state-of-the-art performance.

Acknowledgements. This work was supported by the Key-Areas Research and Development Program of Guangdong Province (No. 2020B010165001). This work was also supported by National Natural Science Foundation of China under Grants 61772527, 61976210, 62076235, and 62002356.

References

1. Bertinetto, L., Henriques, J.F., Torr, P., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. In: ICLR (2019)
2. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: ICCV (2019)
3. Bhat, G., Johnander, J., Danelljan, M., Shahbaz Khan, F., Felsberg, M.: Unveiling the power of deep tracking. In: ECCV (2018)
4. Choi, J., Kwon, J., Lee, K.M.: Deep meta learning for real-time target-aware visual tracking. In: ICCV (2019)
5. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: accurate tracking by overlap maximization. In: CVPR (2019)
6. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: Eco: efficient convolution operators for tracking. In: CVPR (2017)
7. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: ICCV (2015)
8. Fan, H., et al.: Lasot: a high-quality benchmark for large-scale single object tracking. In: CVPR (2019)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
10. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. TPAMI **37**, 583–596 (2014)
11. Huang, L., Zhao, X., Huang, K.: Got-10k: a large high-diversity benchmark for generic object tracking in the wild. TPAMI **43**, 1562–1577 (2019)
12. Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y.: Acquisition of localization confidence for accurate object detection. In: ECCV (2018)
13. Galoogahi, H.K., Fagg, A., Huang, C., Ramanan, D., Lucey, S.: Need for speed: a benchmark for higher frame rate object tracking. In: ICCV (2017)
14. Galoogahi, H.K., Fagg, A., Lucey, S.: Learning background-aware correlation filters for visual tracking. In: ICCV (2017)
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
16. Kristan, M., Leonardis, A., et al.: The visual object tracking vot2018 challenge results. In: ECCV (2018)
17. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: evolution of siamese visual tracking with very deep networks. In: CVPR (2019)
18. Li, F., Tian, C., Zuo, W., Zhang, L., Yang, M.H.: Learning spatial-temporal regularized correlation filters for visual tracking. In: CVPR (2018)
19. Li, X., Ma, C., Wu, B., He, Z., Yang, M.H.: Target-aware deep tracking. In: CVPR (2019)
20. Lin, T.Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
21. Lu, X., Ma, C., Ni, B., Yang, X., Reid, I., Yang, M.H.: Deep regression tracking with shrinkage loss. In: ECCV (2018)

22. Lukezic, A., Vojir, T., Cehovin Zajc, L., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. In: CVPR (2017)
23. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: ICCV (2015)
24. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: a large-scale dataset and benchmark for object tracking in the wild. In: ECCV (2018)
25. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: CVPR (2016)
26. Song, Y., et al.: Vital: visual tracking via adversarial learning. In: CVPR (2018)
27. Tang, M., Yu, B., Zhang, F., Wang, J.: High-speed tracking with multi-kernel correlation filters. In: CVPR (2018)
28. Tang, M., Zheng, L., Yu, B., Wang, J.: Fast kernelized correlation filter without boundary effect. In: WACV (2021)
29. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.: End-to-end representation learning for correlation filter based tracking. In: CVPR (2017)
30. Voigtlaender, P., Luiten, J., Torr, P.H., Leibe, B.: Siam r-cnn: visual tracking by re-detection. In: CVPR (2020)
31. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. TPAMI **37**, 1834–1848 (2015)
32. Zheng, L., Chen, Y., Tang, M., Wang, J., Lu, H.: Siamese deformable cross-correlation network for real-time visual tracking. Neurocomputing **401**, 36–47 (2020)
33. Zheng, L., Tang, L., Wang, J.: Learning robust gaussian process regression for visual tracking. In: IJCAI (2018)
34. Zheng, L., Tang, M., Chen, Y., Wang, J., Lu, H.: Fast-deepkcf without boundary effect. In: ICCV (2019)
35. Zheng, L., Tang, M., Chen, Y., Wang, J., Lu, H.: Learning feature embeddings for discriminant model based tracking. In: ECCV (2020)