A novel approach for visual tracking based on occlusion recognition

Donghai Luo [a], Daobo Wang [b], Shengji Xia [c,] Tingting Bai [d]

College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China

[a]nuaaluo@126.com, [b]dbwangpe@nuaa.edu.cn, [c]18856307679@163.com, [d]baitingting@nuaa.edu.cn

Abstract. Visual object tracking is an extremely challenging task. Many existing trackers cannot handle various challenges simultaneously. In this paper, we propose a novel tracking framework based on an occlusion recognition mechanism to improve the performance in occlusion situations. Firstly, we design an occlusion recognition mechanism based on patch pool and local correlation to describe the occlusion of objects in each frame of an image sequence. Secondly, taking advantage of the occlusion recognition mechanism, we construct a specific training set to train the filter. Thirdly, combining global correlation, we implement our own tracker based on the traditional discriminative correlation filters. Finally, we evaluate it on both OTB and VOT platforms, and the experimental results demonstrate that our design is advanced and effective.

Keywords: Visual tracking, Correlation filter, Occlusion recognition, Local-global correlation.

1. Introduction

Visual tracking is a very fundmental problem in computer vision, which plays an essential role in intelligent surveillance, activity analysis, and autonomous driving [16]. The task of visual tracking is to estimate the trajectory of a given target in a video or a consecutive image sequence, given only its location in the first frame.An ideal tracker should be accurate and robust enough to overcome the unpredictable challenges in a tracking process, such as illumination variation, motion blur, background clutter, scale variation and occlusion.

In the past decade, Discriminative Correlation Filter (DCF) based methods have shown excellent accuracy and robustness, and most of them achieved great real-time performance. A DCF tracker aims to discriminate the target area from the background by training a classifier and update it online. Bolme et al. [4] proposed an adaptive correlation filer (MOSSE) by minimizing the output sum of squared error. Then it was extened and improved by Henriques et al. [14, 15], using circulant data, multi-channel features and nonlinear kernel. Afterwards, its development was drivien by using sophisticated learning models [6, 8, 11], multi-dimensional features [9, 26], and scale estimation [2, 7, 20].

With the development of CNN-structured neural networks, some trackers utilizing deep features have achieved excellent success, such as C-COT [10] and ECO [5]. In addition, some network-based trackers have also appeared. CFNN [19] presents a correlation filter network architecture and a complete tracking pipeline, gaining a promising result. Its biggest feature is that it's updated online and without pre-training. With the advent of gradient backpropagation for DCF [3], another genre emerged, such as DCFNet [24] and DCFNet++ [22]. The former provides a computational basis for the gradients of the DCF tracker, allowing the latter to incorporate the tracker into the training of the feature extraction network. After that, both UDT [23] and self-SDCT [27] use pseudo-labels to integrate forward tracking and backward tracking into the training process, which further improves the training effect of the features network and makes it possible to train on unlabeled datasets.

However, few studies have focused on occlusion, which is a very common problem. To this end, we try to design an occlusion recognition mechanism to obtain the occlusion information of the tracked target, and reduce the influence of occlusion through our tracking framework to improve the performance of the tracker.

2. Our Approach

In order to judge whether the object is occluded, how severe the occlusion is, and which parts are occluded, we propose an occlusion recognition mechanism and use these information to train our filter. When occlusion happens, some parts are hidden behind the background information, thus not every part can be checked in the current frame. In such situations, the object's information is not complete. To keep tracking on the occluded object, an ideal tracker has to make use of its local information. Therefore, following LGCF [28], we introduce local-global correlation into our approach. Fig. 1 provides an overview of our approach. Firstly, we use the DCFNet network to extract deep features. Then global correlation is carried on to determine the scale variations. After that, sampling is performed to extract local information and send the local information to Occlusion Recognition (OR). OR refers to patch pool and generate masks. Finally, according to the masks, the global DCF filter is updated, and then the local correlation is carried on to fix the target's location.

2.1 Review of DCF

Conventional discriminant correlation filters focus on a ridge regression problem, which leads to a closed-form solution. The goal is to find a filter $f(x)=w^T x$ that minimizes loss:

$$\epsilon = \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2 \tag{1}$$

Where x_i denotes the i-th sample, and y_i is the corresponding regression target, usually a gaussian function peaked at the center, and λ is a regularization parameter which controls overfitting. Given by [21], (1) has a closed-form solution:

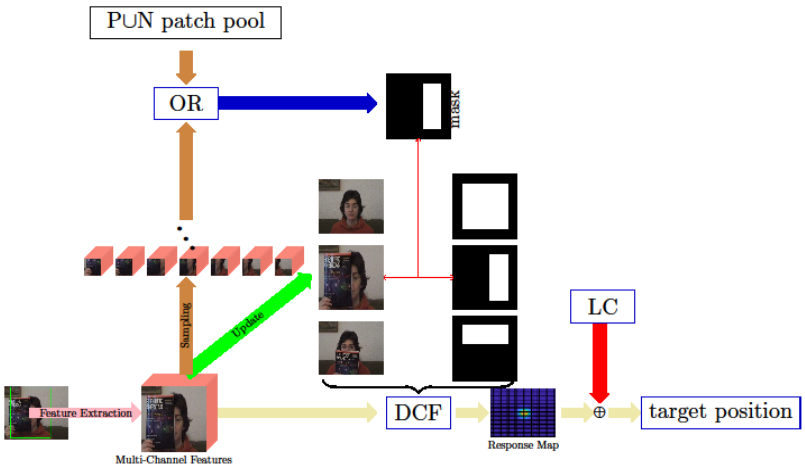

Figure 1. Overview of our approach.

$$w=(X^H X+\lambda I)^{-1} X^H y \tag{2}$$

where X is the data matrix, x_i is the i-th row element, and each element of y is a regression target y_i, I is an identity matrix, H denotes the Hermitian transpose, i.e., $X^H=(X^*)^T$ and X^* is the complex-conjugate of X. Using circulant data X, (2) can be simplified further:

$$\hat{w} = \frac{\hat{x}^* \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda} \tag{3}$$

where $\hat{x}=\mathcal{F}(x)$, $\odot$ denotes the element-wise product, and the fraction is an element-wise division.

The KCF [15] tracker introduce the kernel trick into correlation filters, using high dimensional features $\varphi(x)$:

$$\begin{cases} k(z,x) = \varphi^T(z)\varphi(x) \\ \hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx}+\lambda} \\ \hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha} \end{cases} \tag{4}$$

For a linear kernel, the multi-sample trained filter can be obtained as:

$$\hat{w} = \frac{\sum_i \hat{x}_i^* \odot \hat{y}}{\sum_i \hat{x}_i^* \odot \hat{x}_i + \lambda} \tag{5}$$

2.2 Local Correlation

When occlusion happens, some parts of the target are blocked and cannot be seen. The conventional global-based correlation may degrade due to partial changes. However, local correlation, focusing on the local information rather than the object's overall information, can maintain a good tracking performance. Assuming that the search patch is divided into K parts, our goal is to learn K filters, each of which minimizes the corresponding loss:

$$\epsilon' = (f(x) - y)^2 + \lambda \|w\|^2 \tag{6}$$

The only difference between (6) and (1) is that the former is based on a single base sample, and the latter is based on multiple base samples. In our approach, the loss of (1) is used to describe different occlusion conditions so that the global correlation can keep tracking. While some failures are allowed in local correlation. What must be emphasized is that the reason why DCF can achieve good tracking results is that it acts on a continuous image sequence, and the scene changes (including shape and energy) of the image are not drastic. In local correlation, both shape and energy vary intensely from each other, and the linear kernel almost fails. Compared with linear kernel, gaussian kernel is of stronger expression and distinguish ability. Here, we adopt the gaussian kernel [15] for local correlation.

$$k^{xx'} = \exp\left(-\frac{1}{\sigma^2}\| x - x' \|^2\right) \tag{7}$$

In most cases, the tracked object can be seen as a rigid body, i.e., the relative distance between any two parts remains unchanged. By local correlation, the position correction amount of the target can be gained as:

$$\Delta P = \sum_{i=1}^{j} \mu_i r_i \tag{8}$$

where μ_i is a weight and r_i is a correction calculated by one of the j fine-matching pairs. The calculation of μ_i, r_i and the selection of the pairs would be introduced later

2.3 Occlusion Recognition

Figure 2. Comparison of the training set of the general method and our method.

Inspired by [1] and [13], we establish a patch pool to store local information, both foreground and background, i.e., positive and negative. The pool is obtained by sampling from

the initial frame. Following [13], we take the overlapped patch sampling to gain more spatial information.

In our occlusion recognition scheme, each frame is uniformly sampled and gets N patches. Afterwards, their most similar patches can be found separately in the pool via local correlation. For a patch i, sampled from the current frame, its most similiar patch in the pool is determined as:

$$SP_i = \arg\max_j (\max f_j(p_i)), j = 1,2,\ldots,N \tag{9}$$

where p_i is the search patch in current frame, f_j is the filter based on the j-th element in the pool. So that we could get that patch i in the current frame is close to element SP_i in the pool. And if SP_i includes the information in the foreground (SP_i 's label is positive), patch i is foreground, vice versa. However, the deformation in appearance and the generation of new content information often occurs, and other information may also enter the field of view. Therefore, we consider setting a threshold ρ, and only the pair whose confidence score is greater than ρ is a credible pair, while the others are considered as new information or the uncertain. The confidence cf_i is calculated as:

$$cf_i = \max f_{SP_i}(p_i) \tag{10}$$

Let lab_i denotes the information label of current frame's i-th patch.

$$lab_i = \begin{cases} 1, cf_i > \rho \text{ and } SP_i > 0 \\ 0, cf_i \leqslant \rho \\ -1, cf_i > \rho \text{ and } SP_i < 0 \end{cases} \tag{11}$$

Let vector $\gamma_t^i = [w_t^i \quad h_t^i]^T$ denote the location of patch p_i in current frame t, $\gamma_0^j = [w_0^j \quad h_0^j]^T$ denote the location of sample patch-j in the pool. $f_j(p_i)$, a 2-D matrix, is the correlation response of p_i and sample patch-j. Introduce vector h_j^i:

$$\begin{aligned} h_j^i &= [u \quad v]^T \\ s.t. \quad f_j(p_i)_{(u,v)} &= max f_j(p_i) \end{aligned} \tag{12}$$

And r_i in (8) can be calculated as:

$$r_i = \gamma_t^i + h_j^i - 1 - \gamma_0^j \tag{13}$$

We argue that the background contains information on a variety of other targets, which may not be related to each other. Therefore, we only use the information in the foreground to fix the localization, which means we only select patches with $lab = 1$.

$$\begin{aligned} &\mu_i = cf_i - \rho, i \in \mathbb{O} \\ &\mathbb{O} = \{i \mid lab_i = 1, i = 1,2,\ldots\} \end{aligned} \tag{14}$$

Then we could get a mask matrix $mask_t$ which describes the occlusion of the t-th frame.

$$mask_t = \begin{bmatrix} lab_1 & \ldots & lab_n \\ lab_{n+1} & \ldots & lab_{2n} \\ & \ddots & \vdots \\ & & lab_N \end{bmatrix} \tag{15}$$

2.4 Global Correlation

In order to increase the robustness of the filter and the long-term tracking ability, continuous frames are generally used to train the classifier, but the differences between consecutive frames are very small, which will lead to overfitting and the long-term tracking ability is difficult to guarantee. Inspired by [5], we introduce samples of different occlusion situations into the training set to reduce

the risk of overfitting and improve the performance against occlusion. Fig.2 provides an illustration of this. The general method uses consecutive samples to train the filter (the bottom row), while our method (the top row) selects samples in different occlusion situations to train the filter. We select some representative samples as our training set according to the occlusion of each frame. Our method not only obtains richer and more diverse information, but also makes it possible for the filter to gain performance against occlusion. To select the most characteristic and representative training set, we select the samples with largest $mask$ difference, defined as bellow:

$$d_{ij} = \| mask_i - mask_j \|_2 \tag{16}$$

Besides, we rewrite the loss in (1), assign weights to different samples. The significance of assigning weights is to emphasize the influence of samples that appear more frequently.

$$\epsilon'' = \sum_i \beta_i (f(x_i) - y_i)^2 + \lambda \|w\|^2, \beta_i \in \mathbb{Q} \tag{17}$$

There must be an integer K such that any $\beta_i \times K$ is an integer. Multiply both sides by K:

$$\begin{aligned} K\epsilon'' &= \sum_i K\beta_i (f(x_i) - y_i)^2 + K\lambda \| w\|^2 \\ &= \sum_i B_i (f(x_i) - y_i)^2 + K\lambda \| w\|^2, B_i \in \square \end{aligned} \tag{18}$$

where $B_i = \beta_i \times K$. Then, expand (18) as:

$$K\epsilon'' = \sum_i \sum_j^{B_i} (f(x_{i,j}) - y_i)^2 + K\lambda \| w\|^2 \tag{19}$$

where $(f(x_{i,a}) - y_i)^2 = (f(x_{i,b}) - y_i)^2, a \neq b, i = 1, 2, \ldots$. Going a step further, we can get:

$$K\epsilon'' = \sum_{i,j} (f(x_{i,j}) - y_i)^2 + K\lambda \| w\|^2 \tag{20}$$

By minimizing $K\epsilon''$, our global correlation filter can be obtained:

$$\begin{aligned} \hat{w}' &= \frac{\sum_{i,j} \hat{x}_{i,j}^* \square \hat{y}}{\sum_{i,j} \hat{x}_{i,j}^* \square \hat{x}_{i,j} + K\lambda} = \frac{\sum_i B_i \hat{x}_i^* \square \hat{y}}{\sum_i B_i \hat{x}_i^* \square \hat{x}_i + K\lambda} \\ &= \frac{\sum_i \beta_i \hat{x}_i^* \square \hat{y}}{\sum_i \beta_i \hat{x}_i^* \square \hat{x}_i + \lambda} \end{aligned} \tag{21}$$

2.5 Model Update and Scale Estimation

In global correlation, we build a training set and update the training set with the current frame. Generally, the update strategy can be described as:

$$e_n = (1 - \eta) e_k + \eta e_l \tag{22}$$

where e_l is a new element, e_k is the element to be updated, e_n is the updated element and η is the learning rate. To update the training set, given a new sample x_i, we calculate d defined in (16) between x_i and other samples in the set. Then calculate the distance between the elements in the training set. If the distance between sample x_i and an element in the pool is not more than the distance between any other two elements, we update it with x_i, otherwise we merge the two closest elements and let x_i be a new element in the training set. The process of merging two elements is as follows:

$$\begin{aligned} \pi_n &= \pi_k + \pi_l \\ e_n &= e_k \times (1 - \eta)^{\pi_l} + e_l \times (1 - (1 - \eta)^{\pi_l}) \end{aligned} \tag{23}$$

where π_n denotes the frequency of element e_n.

As for scale estimation, we simply establish a scaling pool $\{t_1, t_2, \ldots, t_k\}$. Assuming that the size of current sample x is (w, h), the size of the k-th sample x^{t_k} in the scaling pool is $(t_k w, t_k h)$, and the optimal scale is calculated as follows.

$$scale = \arg\max_i (\max f(x^{t_i})), i = 1, 2 \ldots, k \tag{24}$$

3. Experimets

Our approach is validated on three benchmarks: OTB-2013, OTB-2015, and VOT2016.

3.1 Implementation Details

The approach proposed in this paper is implemented in Matlab. We apply the DCFNet [24] as our feature extraction network. The regulation parameter λ in (17) is set to 10-4. The influence weights in (17) is set as $\beta = \{8/15, 4/15, 2/15, 1/15\}$. The threshold ρ in (11) is set to 0.3. The patches are 7×7 sampled around the target center, and the size of positive pool is 25 while the negative's is 24. The learning rate η in (22) is set to 0.0105. The σ in gaussian kernel for local correlation is set to 10. As for scale estimation, the scale pool size is 3 and the scale factor is 1.015, namely the pool is set to {1/1.015, 1, 1.015}. In order to save the real-time performance of the algorithm as much as possible, we choose to update the patch pool every 10 frames. A very naive implementation is available at https://github.com/robinluodh/CDCF2022.

3.2 Experiments on OTB-2013

The OTB-2013 dataset [25] is one of the most challenging and convincing datasets in the field of visual tracking. It contains 51 video sequences annotated with target positions, and also annotated the attributes of the targets in each video. There are 11 challenging factors including illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out of view (OV), background clutter (BC) and low resolution (LR). Annotations of these videos and challenges can help researchers analyze the performance of tracking algorithms from different perspectives. OTB-2013 utilizes center location error (CLE) and overlap ratio (VOR) to evaluate trackers' performance.

We evaluated our tracker by one-shot testing (OPE) on the dataset. Fig.3 shows the precision and success plots, with threshold and AUC scores in the figure legend. By comparison, our tracker performs better than some classic trackers, such as CSK [14], Struck [12] and TLD [17], under both criteria. Meanwhile, under the VOR criterion, our tracker performance is comparable to DSST [7], but better under the CLE criterion. Afterwards, it is worth mentioning that another tracker, oursCN, using the Color Name features instead, also performs well, with a drop of only 6%, indicating that our tracker doesn't rely too much on neural network features as others. As mentioned earlier, OTB-2013 also annotated the attributes of each video for us. Below is the performance of our tracker against several specific attributes, as shown in Fig.4. As can be seen from the figure, our tracker has achieved good results in the case of illumination variation, motion blur, and background clutter, but it does not perform well at low resolutions. This may be because we introduce the local correlation branch, so it requires more information.

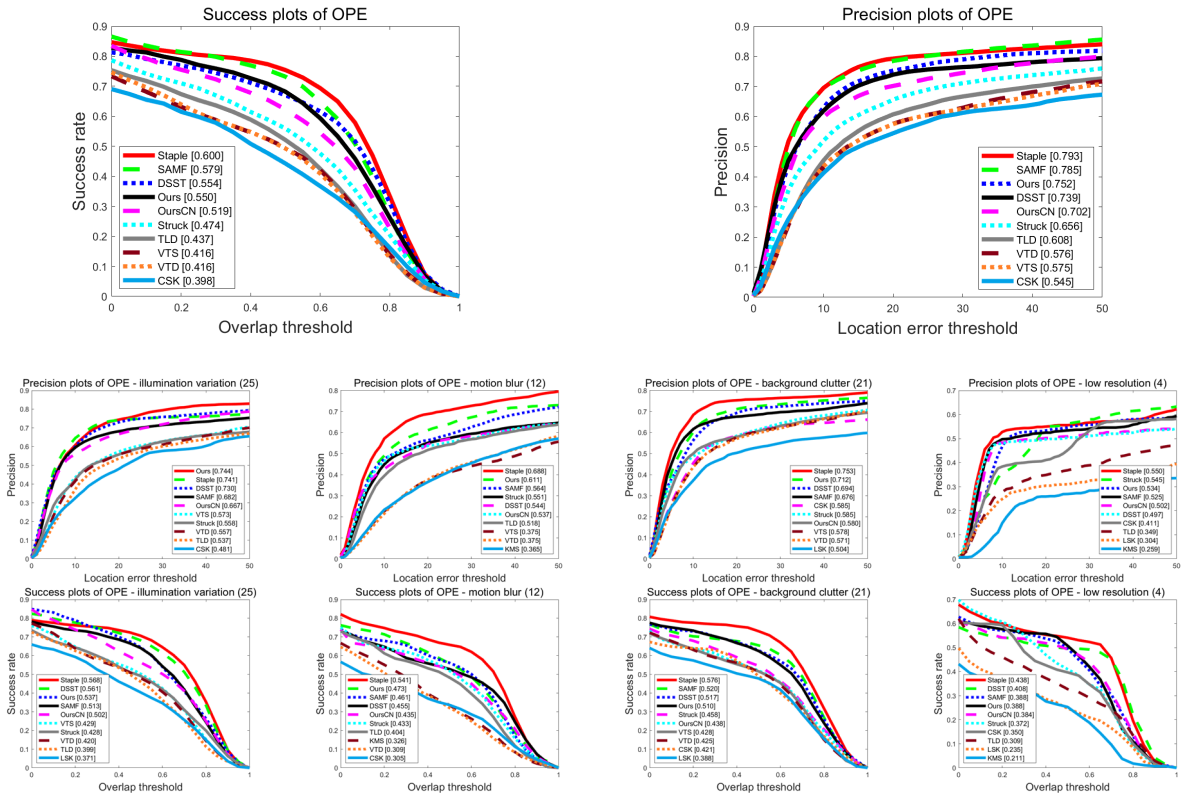

Figure 3. Overall evaluation results on OTB-2013.Evaluation results against IV, MB, BC and LR.

3.3 Experiments on OTB-2015

The OTB-2015 dataset is an extension of the OTB-2013 dataset. On the basis of OTB-2013, 49 video sequences with target positions are added, and the attributes of the targets in each video are also marked. The detailed experimental data on this dataset are shown in TABLE I. Our tracker isn't the best here, but it's still very good, very close to SAMF [26], and even better in some areas. As can be seen from the data, it is far more superior than its baseline, KCF, in all aspects. This fully illustrates the effectiveness and superiority of our designed framework.

3.4 Experiments on VOT-2016

The VOT [18] challenge is one of the most influential annual events. Fig.5 shows the accuracy and robustness of our tracker. From the above data we can see that our tracker performs well. In the A-R plot, it is very close to CCOT, a top tracker in VOT2016, especially under occlusion. Compared with another extremely good algorithm, Staple [2], our algorithm is only slightly less accurate overall and has about the same robustness. When only considering occlusion, our algorithm is much more robust than Staple. The experimental data fully demonstrate the effectiveness of our framework.

4. Summary

We have demonstrated how our design framework works and how we can exploit both global and local correlations to improve tracker performance. We built our own tracker based on traditional DCF tracker and tested it on OTB and VOT, two very influential datasets in the field of object tracking. Experimental results show that although our tracker may not be state-of-the-art at present, it is a large improvement, which fully demonstrates that our design is advanced and effective. Its biggest disadvantage is that the real-time performance is not good enough, and improvement will be the focus of our future work. Its greatest potential is that it is just a framework, and it has the potential to be applied to other better platforms.

Table 2. OPE Results on OTB-2015

Attribute	Metric	Tracker			
		KCF	***DSST***	***SAMF***	***Ours***
FM	***CLE***	0.6214	0.5518	0.6545	0.6395
	VOR	0.4590	0.4468	0.5066	0.5025
BC	***CLE***	0.7132	0.7035	0.6891	0.7113
	VOR	0.4977	0.5227	0.5248	0.5323
MB	***CLE***	0.6005	0.5669	0.6549	0.6528
	VOR	0.4586	0.4688	0.5255	0.5186
DEF	***CLE***	0.6185	0.5328	0.6797	0.6445
	VOR	0.4379	0.4144	0.5048	0.4708
IV	***CLE***	0.7237	0.7155	0.7081	0.7896
	VOR	0.4823	0.5562	0.5304	0.5719
IPR	***CLE***	0.7011	0.6907	0.7206	0.7502
	VOR	0.4694	0.5022	0.5187	0.5400
LR	***CLE***	0.5602	0.5669	0.6845	0.7370
	VOR	0.3074	0.3833	0.4299	0.5071
OCC	***CLE***	0.6317	0.5892	0.7215	0.6802
	VOR	0.4447	0.4490	0.5375	0.5041
OPR	***CLE***	0.6766	0.6442	0.7386	0.7490
	VOR	0.4531	0.4698	0.5363	0.5380
OV	***CLE***	0.5007	0.4806	0.6275	0.6034
	VOR	0.3933	0.3859	0.4803	0.4735
SV	***CLE***	0.6350	0.6332	0.7012	0.7201
	VOR	0.3947	0.4656	0.4921	0.5323
ALL	***CLE***	0.6957	0.6795	0.7513	0.7348
	VOR	0.4771	0.5127	0.5532	0.5436

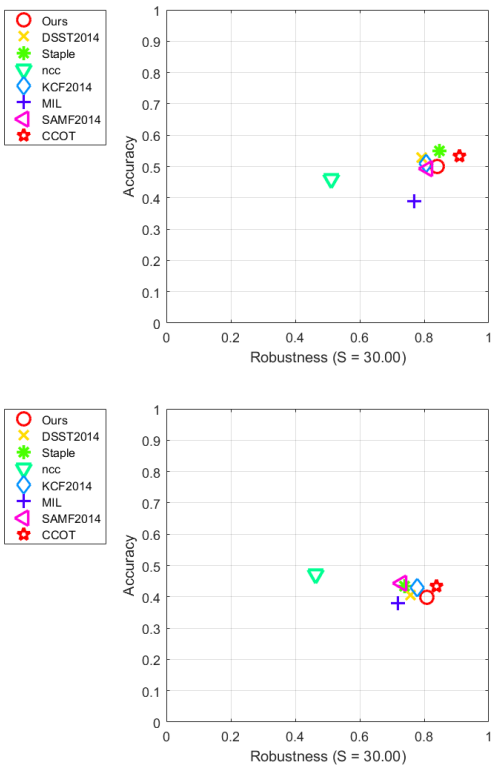

Figure 4. Accuracy-Robustness (A-R) plot (top), A-R plot against occlusion (bottom).

References

[1] B. Babenko, M. Yang and S. Belongie, "Visual tracking with online Multiple Instance Learning," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 983-990.

[2] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik and P. H. S. Torr, "Staple: Complementary Learners for Real-Time Tracking," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1401-1409.

[3] Boeddeker, C., Hanebrink, P., Drude, L., Heymann, J., and Haeb-Umbach, R., "On the computation of complex-valued gradients with application to statistically optimum beamforming." arXiv preprint arXiv:1701.00392 (2017).

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper and Y. M. Lui, "Visual object tracking using adaptive correlation filters," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544-2550.

[5] M. Danelljan, G. Bhat, F. S. Khan and M. Felsberg, "ECO: Efficient Convolution Operators for Tracking," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6931-6939.

[6] M. Danelljan, G. Häger, F. S. Khan and M. Felsberg, "Learning Spatially Regularized Correlation Filters for Visual Tracking," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4310-4318.

[7] M. Danelljan, G. Häger, F. S. Khan and M. Felsberg, "Accurate scale estimation for robust visual tracking," British Machine Vision Conference, Nottingham, September 1-5, 2014. Bmva Press, 2014.

[8] M. Danelljan, G. Häger, F. S. Khan and M. Felsberg, "Adaptive Decontamination of the Training Set: A Unified Formulation for Discriminative Visual Tracking," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1430-1438.

[9] M. Danelljan, F. S. Khan, M. Felsberg and J. Van De Weijer, "Adaptive Color Attributes for Real-Time Visual Tracking," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1090-1097.

[10] Danelljan, M., Robinson, A., Shahbaz Khan, F., & Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," In European conference on computer vision (pp. 472-488). Springer, Cham.

[11] H. K. Galoogahi, A. Fagg and S. Lucey, "Learning Background-Aware Correlation Filters for Visual Tracking," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1144-1152.

[12] S. Hare, A. Saffari and P. H. S. Torr, "Struck: Structured output tracking with kernels," 2011 International Conference on Computer Vision, 2011, pp. 263-270.

[13] Z. He, S. Yi, Y. -M. Cheung, X. You and Y. Y. Tang, "Robust Object Tracking via Key Patch Sparse Representation," in IEEE Transactions on Cybernetics, vol. 47, no. 2, pp. 354-364, Feb. 2017.

[14] Henriques, J. F., Caseiro, R., Martins, P., & Batista, J, "Exploiting the circulant structure of tracking-by-detection with kernels, " In European conference on computer vision (pp. 702-715). Springer, Berlin, Heidelberg.

[15] J. F. Henriques, R. Caseiro, P. Martins and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583-596, 1 March 2015.

[16] Javed, S., Danelljan, M., Khan, F. S., Khan, M. H., Felsberg, M., & Matas, J, "Visual Object Tracking with Discriminative Filters and Siamese Networks: A Survey and Outlook," arXiv preprint arXiv:2112.02838 (2021).

[17] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409-1422, July 2012.

[18] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, Roman P., Gustavo F., et al., "A Novel Performance Evaluation Methodology for Single-Target Trackers," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 11, pp. 2137-2155, 1 Nov. 2016.

[19] Li, Yang, Zhan Xu, and Jianke Zhu. "CFNN: Correlation Filter Neural Network for Visual Object Tracking," IJCAI. 2017.

[20] Ma, Haoyi, Zongli Lin, and Scott T. Acton, "FAST: Fast and accurate scale estimation for tracking," IEEE Signal Processing Letters 27 (2019): 161-165.

[21] Rifkin, Ryan, Gene Yeo, and Tomaso Poggio, "Regularized least-squares classification," Nato Science Series Sub Series III Computer and Systems Sciences 190 (2003): 131-154.

[22] Tian, L., Huang, P., Lin, Z., & Lv, T, "DCFNet++: More Advanced Correlation Filters Network for Real-Time Object Tracking, " IEEE Sensors Journal, 21(10), 11329-11338.

[23] Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., & Li, H, "Unsupervised deep tracking," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[24] Wang, Q., Gao, J., Xing, J., Zhang, M., & Hu, W, "Dcfnet: Discriminant correlation filters network for visual tracking," arXiv preprint arXiv:1704.04057 (2017).

[25] Wu, Yi, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.

[26] Li, Yang, and Jianke Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," European conference on computer vision. Springer, Cham, 2014.

[27] Yuan, D., Chang, X., Huang, P. Y., Liu, Q., & He, Z, "Self-supervised deep correlation tracking," IEEE Transactions on Image Processing 30 (2020): 976-985.

[28] Fan, Heng, and Jinhai Xiang, "Robust visual tracking via local-global correlation filter," Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 31. No. 1. 2017.