# ROBUST VISUAL OBJECT TRACKING WITH SPATIOTEMPORAL REGULARISATION AND DISCRIMINATIVE OCCLUSION DEFORMATION

*Shiyong Lan*[†], *Jin Li*[†], *Shipeng Sun*[†*], *Xin Lai*[★], *Wenwu Wang*[‡]

[†]College of Computer Science, Sichuan University, China.
[★]School of Mechanical and Electrical Engineering, Southwest Petroleum University, China.
[‡]Center for Vision Speech and Signal Processing, University of Surrey, UK.

## ABSTRACT

Spatiotemporal regularized Discriminative Correlation Filters (DCF) have been proposed recently for visual tracking, achieving state-of-the-art performance. However, the tracking performance of the online learning model used in this kind methods is highly dependent on the quality of the appearance feature of the target, and the target feature appearance could be heavily deformed due to the occlusion by other objects or the variations in their dynamic self-appearance. In this paper, we propose a new approach to mitigate these two kinds of appearance deformation. Firstly, we embed the occlusion perception block into the model update stage, then we adaptively adjust the model update according to the situation of occlusion. Secondly, we use the relatively stable colour statistics to deal with the appearance shape changes in large targets, and compute the histogram response scores as a complementary part of final correlation response. Extensive experiments are performed on four well-known datasets, i.e. OTB100, VOT-2018, UAV123, and TC128. The results show that the proposed approach outperforms the baseline DCF method, especially, on the TC128/UAV123 datasets, with a gain of over 4.05%/2.43% in mean overlap precision. We will release our code at `https://github.com/SYLan2019/STDOD`.

***Index Terms***— Visual Tracking, Discriminative Correlation Filters, Spatially-Temporal Regularization, Occlusion.

## 1. INTRODUCTION

Visual object tracking, as one of the most important foundations of numerous applications of computer vision, has made a rapid breakthrough in recent years. The existing visual object tracking methods can be divided into generative and discriminative methods. The generative methods aim at describing the target appearance, using some generative processes, e.g., template matching or statistical learning, and then searching for the candidate objects to minimize reconstruction errors. In contrast, the discriminative methods regard target tracking as a classification problem by distinguishing the target appearance from its background, such as Discriminative Correlation Filters (DCFs) [1, 2, 3, 4], and Siamese networks [5].

The DCFs have several advantages. First, as a type of regression-based trackers, they can directly learn mapping from dense samples around the target to Gaussian-like soft labels, and do not need to manually annotate a large number of video or image data. Second, with the circular shifts of training samples, the convolution operation used in correlation can be converted to Fourier domain to achieve high computational efficiency. Third, with robust multi-feature representation for target [4, 6], such as hand-crafted HOG, Color Names (CN) [7] features and deep convolutional neural network (CNN) features [6], the tracking accuracy and robustness of the DCFs can be further improved. Finally, spatiotemporal regularisation has been used in the DCF trackers to improve tracking performance [3], for example, spatial attention based mechanism has been incorporated into the online learning model to improve its ability in discriminating the target from the background clutters [8], and temporal continuity of successive frames has been exploited to learn a more robust target appearance model [3, 4].

However, the performance of DCF methods, such as GFS-DCF [4], STRCF [3], HCFstar [6], degrades significantly when the target is deformed between successive frames, and in this scenario, the response of the x-correlation between the model and the candidate targets will be distorted.

To address this problem, patch-based trackers such as [9] exploit correlation filters to track separate parts of the entire target, and then adopts an adaptive strategy to combine the confidence maps from all parts. However, these patch-based trackers cannot discriminate whether the deformation is caused by occlusions by other objects or shape changes by the target itself. Different types of deformation may require different control strategies for model update. For example, the target model should be updated with the target appearance in current frame in order to capture target's own deformation,

*Corresponding author.

This work has been funded partially by the key R&D project (2021YFG0300) of the Sichuan Science and Technology Department, China, and by the National Defence Key Laboratory of Synthetic Vision Graphics and Images, Sichuan University, China.

while in the presence of heavy occlusions, the target model should not be updated.

In this paper, we focus on determining whether the target deformation is caused by object occlusion or self-appearance change. We propose a new visual object tracking method where we take into account the difference between occlusion and self-deformation in a spatiotemporal regularised DCF filter. We term this new method as STDOD-DCF. The main contributions can be summarized as follows.

(1) We propose a new method based on a spatiotemporal regularised DCF for robust visual tracking, which considers the difference between occlusion-induced deformation and target self-deformation. More specifically, we use four detectors, initialized in the adjacent regions outside the target boundary in four directions, i.e. up, left, bottom, and right, respectively, and then automatically detect whether the region outside the target enters into the region of the target.

(2) We propose an adaptive strategy for model update according to the difference between occlusion and self-deformation of target. The proposed strategy can not only enhance the tracking performance with rapid target self-deformation, but also reduce tracking drift caused by occlusion.

(3) We perform extensive experiments to evaluate our method on the OTB100, VOT-2018, UAV123, and TC128 datasets, and show that it performs better than most of the state-of-the-art trackers. On TC128/UAV123, our method outperforms the baseline DCF, by over 4.05% and 2.43% respectively, in mean overlap precision.

## 2. OUR APPROACH

In this section, we present the STDOD-DCF tracking framework which contains three main parts as shown in Fig. 1. The first part is the target tracking using a spatiotemporal regularisation based DCF method. The second part is used to predict the location of the patches surrounding the target in parallel to the DCF method. The third part is the model update strategy by exploiting the difference between occlusion and self-deformation to help avoid the model contamination.

### 2.1. Target tracking

In this part, we construct the main body of our target tracking algorithm by combining two state-of-the-art DCF-based trackers (i.e., STRCF [3] and GFS-DCF [4]). Thanks to the spatiotemporal regularisation, the STRCF is not only effective in suppressing the adverse boundary effects with spatial regularisation, but also efficient in computation by exploiting temporal regularisation to rationally approximate multiple training samples with single one.

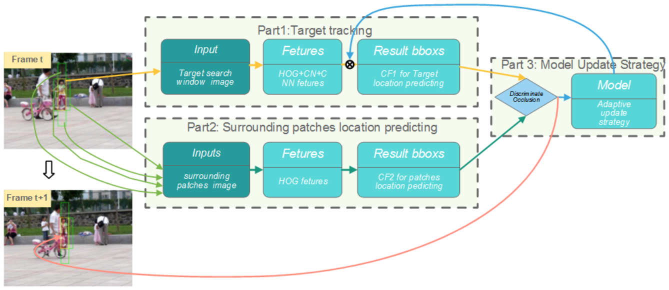
Frame t
Frame t+1
Part1:Target tracking
Input
Target search window image
Fetures
HOG+CN+C NN fetures
Result bboxs
CF1 for Target location predicting
Part2: Surrounding patches location predicting
Inputs
surrounding patches image
Fetures
HOG fetures
Result bboxs
CF2 for patches location pedicting
Part 3: Model Update Strategy
Discriminate Occlusion
Model
Adaptive update strategy

**Fig. 1**. The proposed method consists of three parts: target tracking, prediction of the location of the patches surrounding a target, and an adaptive update strategy based on target deformation information.

Specificly, in order to distinguish the target from the background, based on the consideration of the training pair $\{X_t, Y_t\}$ in frame $t$, where $X_t$ is the multi-channel features, $Y_t$ is its corresponding predefined Gaussian shaped label, the STRCF formulates the objective as a regularised least squares problem which learns the multi-channel discriminative filters $\widetilde{F}_t$.

$$\widetilde{F}_t = \arg\min_{F} \| \sum_{j=1}^{C} {F_t}^j \circledast {X_t}^j - Y_t\|^2 + \frac{1}{2}\sum_{j=1}^{C} \|W^j \cdot {F_t}^j\|^2 + \frac{\mu}{2}\|F_t - F_{(t-1)}\|^2, \quad (1)$$

where $\circledast$ stands for the convolution operator, $\cdot$ denotes the Hadamard product, ${X_t}^j \in \mathbb{R}^{M\times N}$ and ${F_t}^j \in \mathbb{R}^{M\times N}$ are the $j$-th channel feature representation and the corresponding discriminative filter respectively, $M \times N$ denotes the size of $j$-th channel feature map, $C$ is channel number of feature $X_t$, and $W$ and $F_t$ are the spatial regularization matrix and correlation filter, respectively. $\sum_{j=1}^{C} \|W^j \cdot {F_t}^j\|^2$ denotes the spatial regularizer, $W^j$ is the spatial regularization matrix of the $j$-th channel, whose size is $MN \times MN$, $\|F_t - F_{(t-1)}\|^2$ denotes the temporal regularizer which can help the tracker to retain temporal coherence, and $\mu$ is the corresponding penalty factor.

Considering the information redundancy caused by too many channels in deep CNN features, we introduce channel selection regularization term referring to GFS-DCF [4].

$$\widetilde{F}_t = \arg\min_{F} \| \sum_{j=1}^{C} {F_t}^j \circledast {X_t}^j - Y_t\|^2 + \lambda_c \sum_{j=1}^{C} \|{F_t}^j\| + \frac{\lambda_1}{2}\sum_{j=1}^{C} \|W^j \cdot {F_t}^j\|^2 + \frac{\mu}{2}\|F_t - F_{(t-1)}\|^2, \quad (2)$$

where $\lambda_c$ is the regularization parameter for channel selection, the second term in Eq. (2) focuses on channel sparsity to realize feature channel selection, here, the channels with the lowest channel attributes are set to zero by preset proportion.

In learning stage, the model in Eq. (2) is convex, and can be minimized to obtain the globally optimal solution via alternating direction of multiple multipliers (ADMM). A closed-form solution $\widetilde{F}_t$ can be obtained efficiently in the frequency domain [3, 4]. Then, we can use an updating rate $\xi \in [0, 1]$ to control the update of tracking filter $F_{t+1}$ as $\xi\widetilde{F_t}$ +(1-$\xi$)$F_t$ from the $t$-th frame, where the $\xi$ will be adjusted according to the discriminative deformation of the target in Section 2.3. In the tracking stage, we can directly compute the response of the x-correlation between the filter $F_{t+1}$ and input features $X_{(t+1)}$ in the frequency domain, and then locate the target in the $(t + 1)$-th frame by selecting the maximum value in the response map with inverse discrete Fourier transform (DFT).

### 2.2. Surrounding patches location predicting

Occlusion in video normally comes from target's surroundings, so we need to be aware of its context patches in real time. Similar to CACF [10], we initialize four surrounding patches at left, right, top and bottom of the target, respectively, in the first frame.

Then, we track and locate the target and its surrounding patches in parallel, meanwhile we can compute the location relationship between target and its surrounding patches, so as to determine whether the target is occluded by its surrounding patches. If someone surrounding patch locates into the target region from $t$-th to $(t + 1)$-th frame, the target is regarded as being occluded. When a patch occludes the target, we keep tracking this patch before it moves out of the bounding box of the target. Usually, this is a short-term object tracking problem, therefore, the classical KCF [2] can be used for this task for its computational efficiency.

As for each surrounding patch of the target, a KCF tracker is used to predict its location, separately. As a result, we can deal with the location prediction for all surrounding patches in parallel. In addition, we use only one scale for each target's surrounding patch for simplification. Fig. 2 shows the result of the location prediction for each surrounding patch.

### 2.3. Model updating strategy

As discussed above, some slight deformation or trivial clutter from background can be well coped by existing spatiotemporal DCFs (such as GFS-DCF, STRCF). Here, we focus on how to discriminate the occlusion from self-deformation of target and how to adaptively update the model.

First, we determine the confidence of the response map obtained from the surrounding patches in terms of the quality measured with the peak-to-sidelobe ratio (PSR) [11], and compared with a threshold $Th_{SPresp\{i\}}$, for the $i$-th surrounding patch. If the response of a surrounding patch is higher than the threshold, it is regarded as overlapping with the target, i.e. the target is occluded by this surrounding patch. If the area of overlap between the target with all its surrounding patches is greater than a pre-defined threshold $Th_{occ}$, the target is considered overlapped, and an indicator $b_{occ}$ is set as $b_{occ} = 1$, which can be used to adjust $\xi$ and limit the model update.

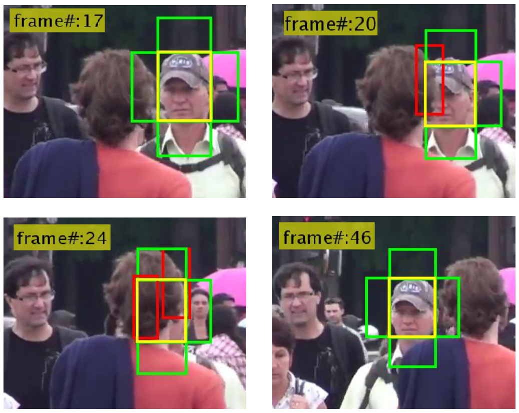

**Fig. 2**. An illustration on how a target is occluded by its surrounding patches. The surrounding patches (*greenbox*) are in the adjacent regions outside the target (*yellow*) boundary. If a patch (*red*) occludes the target, it will be kept in a set of candidate patches and used for location predication in the next frame, otherwise it will be removed from the set of patches and reset new one according the target.

If no occlusion has been detected by the patch-based detectors, but the confidence level of the x-correlation response mapping from the target is low, in terms of a defined quality threshold of $Th_{Tgresp}$ for the target response, then it means that some self-appearance deformation of target is occurring. In this case, we compute the histogram response scores in terms of colour statistics, as a complementary part of the final correlation response. We then adjust $\xi$ according to response quality and update the model to capture the rapid self-deformation of target. To improve the reliability of quality, we also consider another response's status, i.e. the maximal-value-variate-ratio $r_{mv}$ to the previous frame, whose value changes more significantly than that in target self-deformation when the target gets in or out of occlusion. Therefore, with the quality of target response and area of target being occluded, we can quantify the extent that the target is occluded, and thus can achieve more effective adjustment than many existing methods [6, 11].

## 3. EXPERIMENTAL EVALUATIONS

To evaluate the proposed method, we perform experiments on four well-known datasets, i.e. OTB100 [12], TC128 [13], UAV123 [14], and VOT2018 [15], respectively.

Our tracker is implemented in MATLAB 2018a. Its speed is about seven fps on a platform with one CPU (Intel Xeon E5-2637 v3) and one GPU (NVIDIA GeForce-GTX-TITAN-X). We use ResNet-50 deep CNN features, HOG and CN hand-crafted features in our experiments. We set $Th_{SPresp\{i\}} = 1.15 * min(\sqrt{Area_{Patch(i)}}, 200)$ for each patch, $r_{mv} = 1/1.3$, $Th_{occ} = 0.83$, $Th_{Tgresp} = 0.85$, and $\beta = 0.02$. The size of left/right surrounding patches are (W/2)*H, the size of up/bottom patches are W*(H/2), where W*H is the target size. Here, these above parameters are selected by extensive empirical tests. The other parameters for target tracking refer to GFS-DCF [4], and other parameters for surrounding patches location refer to KCF.

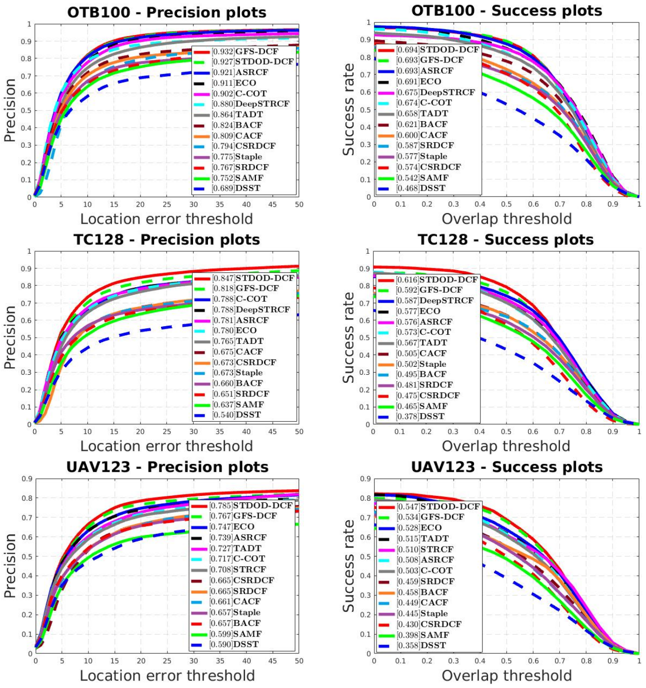

**Fig. 3**. The precision plots (*left*) and success plots (*right*) on OTB100, TC128 and UAV123, where the precision plots show the distance precision (DP) value with a threshold of 20 pixels, and the success plots show the overlap success value with the area under the curve (AUC).

To evaluate the tracking performance, we follow the protocols [16, 15], and compare our tracker with many state-of-the-art trackers (mainly DCF-based trackers), including GFS-DCF [4], ECO [1], CCOT[17], CFCF [18], LADCF [19], Staple [20], STRCF [3], ASRCF [11], SiamFC [5], BACF[21], LSART [22], and some other trackers in VOT2018 challenges. The one-pass evaluation (OPE) is used to evaluate the tracking performance on UAV123, TC128 and OTB100. According to the score shown in the legend in Fig.3, we can see that our method achieves high performance on all the four datasets. Especially, our tracker outperforms the second best baseline, by a significant margin (3.55%, 4.05%) and (2.35%, 2.43%) in term of $(DP, AUC)$ on TC128 and UAV123 respectively.

To verify the robustness of our tracker, we use the average results of unsupervised experimental evaluation on the vot2018 dataset, which contains video sequences with many classification attributes, such as occlusion, scale change, motion change and so on. Moreover, this vot2018 benchmark can evaluate the tracking performance of the classified video sequences separately and comprehensively. Table 1 shows the results on VOT2018. Our algorithm ranks first in term of the overall indicator *tag_all*, which is 3.3% better than the second-best. In particular, the *tag_occlusion* score is improved by 10.4%, *tag_size_change* and *tag_motion_change* are improved by 5.2% and 5.1% respectively compared with the second-best tracker. This experiment demonstrates that our tracker can more effectively overcome the impact of occlusion or scale deformation on visual object tracking, so as to validate the robustness of our algorithm. Fig. 4 shows some tracking examples by the proposed method and the baseline methods in the case of occlusion and self-deformation.

**Table 1**. VOT2018 unsupervised_overlap_average overview.
(The red, blue and green represent the best three results respectively.)

| | *tag_camera_motion* | *tag_empty* | *tag_illum_change* | *tag_motion_change* | *tag_occlusion* | *tag_size_change* | *tag_all* |
|---|---|---|---|---|---|---|---|
| STDOD-DCF | 0.5210 | 0.4136 | 0.4576 | 0.4944 | 0.3560 | 0.4812 | 0.4586 |
| GFS-DCF | 0.4865 | 0.3959 | 0.5007 | 0.4700 | 0.3175 | 0.4578 | 0.4344 |
| DeepSTRCF | 0.4963 | 0.4074 | 0.4105 | 0.4391 | 0.3225 | 0.4301 | 0.4365 |
| LADCF | 0.4897 | 0.3882 | 0.4089 | 0.4575 | 0.3204 | 0.3885 | 0.4213 |
| LSART | 0.4496 | 0.4569 | 0.4316 | 0.4414 | 0.2719 | 0.4016 | 0.4389 |
| ECO | 0.4189 | 0.4132 | 0.4251 | 0.3701 | 0.2804 | 0.3695 | 0.4025 |
| BACF | 0.2759 | 0.2414 | 0.3254 | 0.2548 | 0.2056 | 0.1759 | 0.2447 |
| SiamFC | 0.3598 | 0.3489 | 0.3867 | 0.3358 | 0.2385 | 0.3310 | 0.3428 |
| KCF | 0.2816 | 0.2431 | 0.3202 | 0.2704 | 0.2628 | 0.2774 | 0.2671 |
| MCPF | 0.4504 | 0.4702 | 0.3606 | 0.4244 | 0.2724 | 0.4560 | 0.4440 |
| CCOT | 0.4002 | 0.4113 | 0.3618 | 0.3386 | 0.2767 | 0.3532 | 0.3909 |
| CFCF | 0.4271 | 0.3453 | 0.3575 | 0.3596 | 0.3096 | 0.3534 | 0.3773 |
| Staple | 0.3965 | 0.2880 | 0.3538 | 0.3776 | 0.2275 | 0.3253 | 0.3327 |

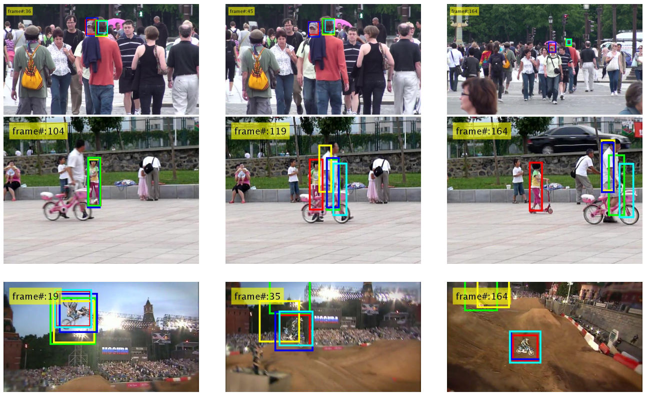

**Fig. 4**. A comparison of our approach (red) with ASRCF (yellow), HCFstar (blue), STRCF (green), and GFS-DCF (cyan) on three sequences (*i.e*, *tc_Face_ce* from TC128, *Girl*2 and *MotorRolling* from OTB100) with heavy occlusion or deformation. The images can be zoomed in for a better view.

## 4. CONCLUSION

We have presented a new DCF method for visual object tracking by discriminating occlusion from self-deformation of the target, and using an adaptive strategy for model update. With extensive experiments, we have demonstrated significantly improved tracking accuracy and robustness compared with several popular baseline DCF trackers.

## 5. REFERENCES

[1] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6638–6646.

[2] J. F. Henriques, C. Rui, P. Martins, and J. Batista, "Highspeedtracking with kernelized correlation ï¬lters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[3] F. Li, C. Tian, W. Zuo, L. Zhang, and M.H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4904–4913.

[4] T. Xu, Z.H. Feng, X.J. Wu, and J. Kittler, "Joint group feature selection and discriminative filter learning for robust visual object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7950–7960.

[5] L. Bertinetto, J. Valmadre, J.F Henriques, A. Vedaldi, and P. HS Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 850–865.

[6] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, "Robust visual tracking via hierarchical convolutional features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, August 2018.

[7] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.

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

[9] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4902–4912.

[10] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1396–1404.

[11] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4670–4679.

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

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

[14] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *European conference on computer vision*. Springer, 2016, pp. 445–461.

[15] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, and et al., "The sixth visual object tracking vot2018 challenge results," *IEEE Conference on European Conference on Computer Vision*, 2018.

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

[17] 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*. Springer, 2016, pp. 472–488.

[18] E. Gundogdu and A A. Alatan, "Good features to correlate for visual tracking," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2526–2540, 2018.

[19] T. Xu, Z.H. Feng, X.J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5596–5609, 2019.

[20] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. HS Torr, "Staple: Complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1401–1409.

[21] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1135–1143.

[22] C. Sun, D. Wang, H. Lu, and M. H. Yang, "Learning spatial-aware regressions for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8962–8970.