FAST ROBUST TRACKING VIA DOUBLE CORRELATION FILTER FORMULATION

Ashwani Kumar Tiwari, Rahul Siripurapu, Yadhunandan U S

Computer Vision Lab, Smart Machines Group, Chief Technology Office,
Wipro Technologies, Bangalore, India

ABSTRACT

Over the past few years, fast and robust trackers based on Kernelized Correlation Filters have shown top notch performance on the Visual Object Tracking challenge. However there is still scope for obtaining higher performance through the use of reasonable approximations that can easily be shown to work through empirical methods. We study some variants derived from the Discriminative Scale Space Tracker and show significant improvement in tracking performance. Our tracker outperforms both fDSST and DSST on the VOT 2016 and 2017 datasets in terms of both Expected Average Overlap (EAO) and Equivalent Filter Operations (EFO). We also demonstrate that the error correcting capability inherent in our method leads to a higher performance on the unsupervised VOT 2016 and 2017 benchmarks.

Index Terms— Object Tracking, Kernelized Correlation Filter, Discriminative Scale Space Tracker

1. INTRODUCTION

Visual object tracking, especially the model free variant using tracking-by-detection has come a long way from the primitive methods based on template matching to correlation filters using multiple high level features. These methods are invariant to most of the commonly occurring phenomena like illumination changes, deformations, occlusions and rotations that make tracking challenging [4]. Over the years, even the tracking benchmarking measures have moved from the naive Distance Precision (DP) [3,6,19], Center Location Error(CLE) [6], Average Overlap (AO) [6,20] and others [8,9,10,34,35,36] to Accuracy, Robustness and EAO, which are proposed in Visual Object Tracking (VOT) challenge [7] (currently the largest and most challenging benchmark for short-term-single-object trackers that do not apply pre-learned object appearance models [7]). VOT also proposed a benchmark for speed performance termed EFO [21].

Most of the state-of-the-art methods, like CCOT [22], TCNN [23], SSAT [7,24], an extension of MDNet [24], MLDF [7,26,27], Staple [15], DNT [28], SSKCF [7,29,4], SiameseFC [14] and DeepSRDCF [7,30], use either some form of multilevel features or deep neural network based features and are hence very slow in operation. The best tracker in terms of EAO in VOT2016 [7] is CCOT, which is based on a Continuous Convolution Operator; it extends DCF [4] to learn multi-resolution deep feature maps by using an interpolation function, thus enabling sub pixel estimates of the object location and it runs at 0.55fps [31]. TCNN uses multiple CNNs to model the object appearance and runs at 1.5fps using an i7-5820K and TITAN X [23]. SSAT, Scale and State aware tracker is an integration of image segmentation into MDNet, which in turn uses a Multi-Domain Network, where separate branches of the neural net are fine tuned for multiple domains (object types) in the tracking dataset and this runs at 1fps on an 8 core XeonE5 along with a TeslaK20m GPU [24]. Multi-Level Deep Feature (MLDF) Tracker uses VGGnet [32] features to train separate networks for localization and scale estimation and is again slow [7]. DNT or Dual Deep Network Tracker also uses pre-trained Deep features and is slow [7]. Such methods, despite their stellar performance on the VOT 2016 benchmark, perform poorly on the VOT 2017 real time challenge where the trackers are run on the sequences with real time frame rates [38]. Here faster trackers have a significant advantage in performance [33].

For real time tracking, speed is very essential and most of the fast methods currently in use are based on the Kernelized Correlation Filter (KCF) framework that exploits the circulant nature of the kernel matrices involved in densely sampled tracking-by-detection algorithms [1,2,3]. It's an approach that has been adapted from the Minimum Output Sum of Squared Error (MOSSE) [1]. The tracking problem is framed as a minimization of the sum of squared error with respect to a gaussian response. This in turn simplifies into a ratio, where the numerator is the correlation of the input with the gaussian and the denominator is the energy spectrum of the input. To adapt this into an online learning framework, a running average is performed on both the terms of the ratio for every frame with a suitable learning rate [2]. Henriques et al. generalize this further by incorporating the Kernel Trick, producing a kernelized correlation filter. Further they use HoG features [13] which provide higher immunity to noise.

Discriminative Scale Space Tracking (DSST) [6] is one approach to incorporate scale tracking into KCF. It is built upon the KCF framework, with the HoG feature being replaced by fHoG features [16]. A separate scale filter, which consists of the target features extracted at 33 different scales is used along with the same KCF approach to track the object in this constructed scale space. Danelljan et al. further improve this in terms of speed and distance precision with the fast DSST (fDSST) [11] by using dimensionality reduction via PCA and sub-grid interpolation in both the scale and spatial filters. Along with this they also extend the search space by using a higher padding. fHoG is applied with 4x4 bin size to reduce the spatial extent of the filter by a factor of 4. Thereafter, PCA is used to reduce the 32 channel fHoG+grayscale image feature to 18 channels. This significantly reduces computation despite the increased search space. Sub-grid interpolation is used to estimate the location at 4x the spatial filter size and scale at 2x the scale filter size (scale estimation at 33 levels is obtained using only 17 scales). One of the motivations for

the proposed work lies in the fact that a separate scale filter is quite expensive computationally and can be avoided.

KCF has been adapted by several other authors as well, such as Scalable Kernel Correlation Filter with Sparse Feature Integration (sKCF)[37], which is an improved KCF tracker that uses a scalable gaussian window and a keypoint based model and is able to overcome the fixed size limitation in KCF. Adaptive Regression Target DSST is an extension of DSST with an anisotropic gaussian regression target that helps it handle oblong objects better than the original DSST [7].

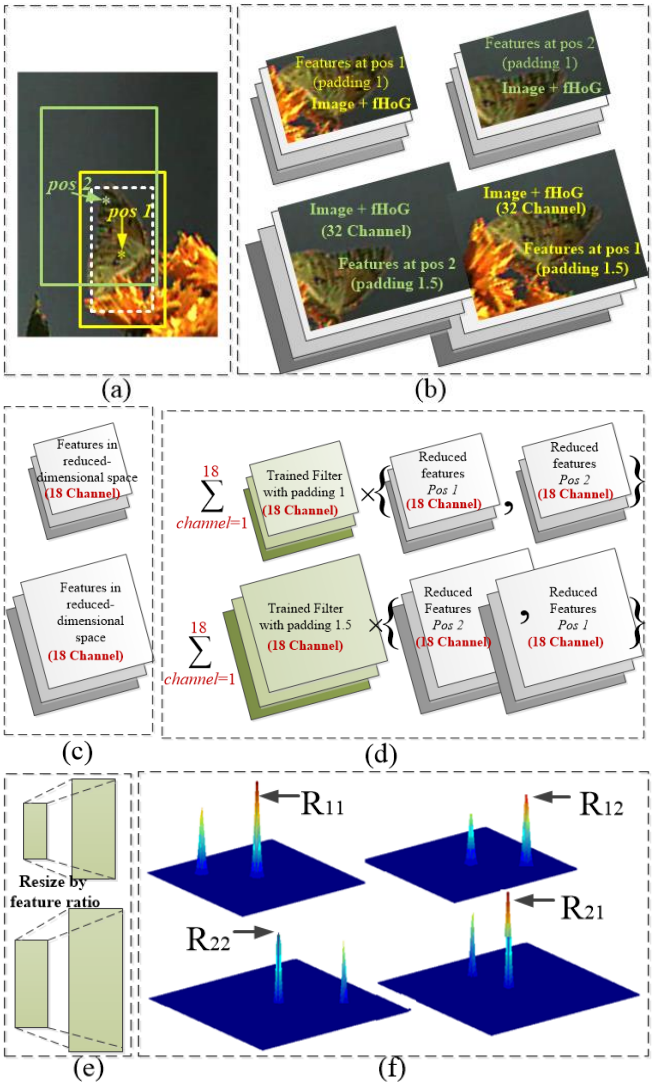

Fig. 1: Proposed error correction algorithm: a) White box: object to be tracked. Yellow box: 1.0 padding patch. Green box: 1.5 padding patch. b) 31 channel fHoG features along with grayscale image feature extracted from patch c) Feature matrices reduced to 18 channels from 32 via PCA. d) Multiplication in Fourier domain e) Response resized to original patch size. f) Responses in spatial domain.

Another approach including scale estimation built upon KCF, Scale Adaptive KCF Tracker with Feature Integration (SAMF) [5] by Li et al computes the filter responses at multiple scales. These trackers are simple, fast and accurate but their performance can be improved upon as done by the proposed tracker which is detailed in the following section.

2. PROPOSED METHODOLOGY

Since most of the above discussed trackers have a low frame rate [7,33], we propose two simple modifications to fDSST, 1) double correlation filters which make the tracker robust to failure and 2) sequential scale estimation that can help handle scale changes efficiently. The first modification which provides error correction makes use of filters with paddings 1 and 1.5, termed K_1 and $K_{1.5}$ respectively. Having a larger padding filter has its pros and cons. It can lead to more background leaking into the filter, making it more likely to lose track in cases with cluttered background, partial occlusions and rapid deformations in the object. Whereas filters with smaller padding tend to lose track of the object during phases of fast movement. Hence using filters of different sizes can be complementary. The two filters are used simultaneously to track the object giving two target location estimates pos1 and pos2 as shown in Fig. 1a. The error correction is achieved by evaluating the filters at both locations at every time step, which results in responses R_{11}, R_{12}, R_{21} and R_{22}. Where R_{11} is the response of filter 1 at pos1, R_{12} is the response of filter 1 at pos2 and so on.

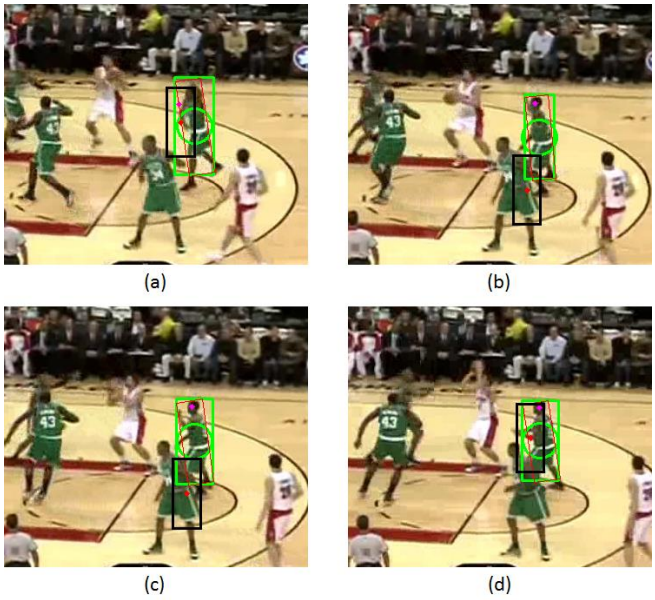

Fig. 2: An example of error correction: The green box is ground-truth and the black box is the proposed tracker. The tracker loses the target while transitioning from 2a to 2b. Consecutively, while transitioning from 2c to 2d it regains the target.

Pseudo codes for train, update and detect are similar to the ones presented in [11]; the error correction methodology is explained in the pseudo code below.

function [pos1, pos2] = Error_correction(R_{11}, R_{12}, R_{21}, R_{22}, pos1, pos2, noise_threshold = 1.05)
    if max(R_{12}(:)) > max(R_{11}(:)) × noise_threshold
    //threshold used to avoid switching due to noise
        pos1 = pos2;
    elseif max(R_{21}(:)) > max(R_{22}(:)) × noise_threshold
        pos2 = pos1;
    end
end

The above discussed algorithm is presented in Fig. 1 where filter $K_{1.5}$ fails while K_1 continues tracking (refer Fig 1a). In the successive frame $K_{1.5}$ corrects using the location estimate of K_1 because R_{21} is higher than R_{22} (refer Fig 1f). An example of this is demonstrated in the Fig. 2.

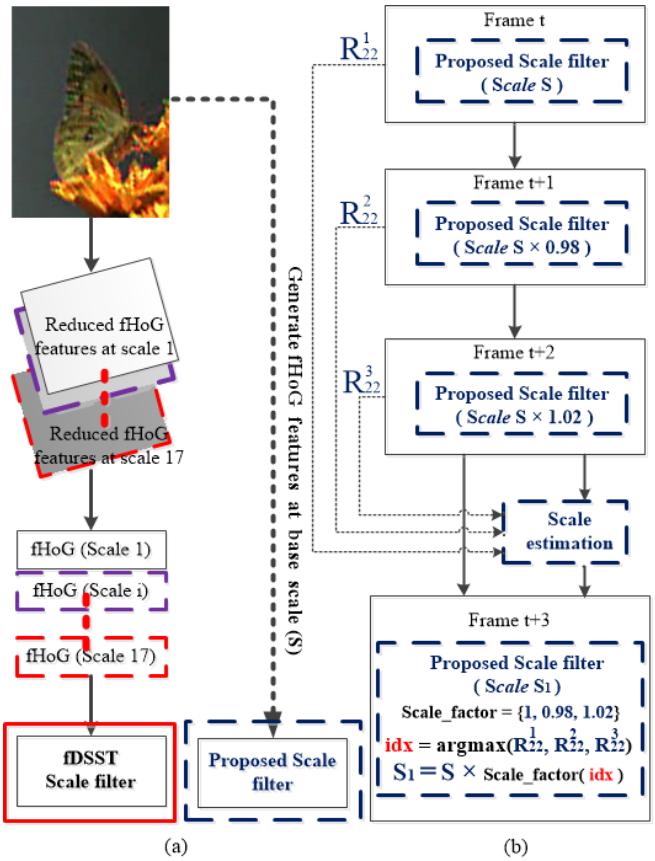

Fig. 3: Proposed methodology for fast scale estimation.

The second modification is the use of a scale factor in {0.98, 1.0, 1.02} for scaling the target area on consecutive frames for the translation filter. The translation filter is then used to estimate scale without using a separate scale filter. The use of separate scale filters in fDSST (Fig. 3a) and SAMF are computationally costly and redundant. In our approach the scale responses are computed in different frames [sequentially] as shown in Fig. 3b. The responses of a filter for each of the three scales (R_{22}^1, R_{22}^2 and R_{22}^3, where superscript indicates the frame index) can be compared across frames under the assumption that changes between consecutive frames are not significant.

The pseudocode for the above method is as follows:
function base_scale = scale_estimation (R_{22}^1, R_{22}^2, R_{22}^3, base_scale, scale_factor)
    idx = argmax ([R_{22}^1, R_{22}^2, R_{22}^3]);
    base_scale = base_scale × scale_factor(**idx**)
end

These two modifications enable us to achieve both higher EAO as well as a higher speed compared to the state-of-the-art as detailed in the next section.

3. EXPERIMENTAL RESULTS AND DISCUSSION

For evaluation we have used the publicly available VOT 2016 and 2017 datasets [7,38] each consisting of 60 videos which exhibit a range of challenging situations. Parameters that we use for benchmarking our proposed tracker are Accuracy, Robustness, EAO and EFO. Accuracy is merely the average overlap in all frames where tracker has not failed. Robustness is the average number of failures per sequence. EAO is the average of the expected overlap curve evaluated over an interval of average short term sequence lengths without resets [8]. EFO is intended to be a system independent measure of tracker speed [7,8]. In the VOT 'baseline' (B) experiment, the tracker is reset upon a failure condition to ensure that the measures are independent of sequence lengths [7]. Since the proposed tracker has an error correct capability, to ascertain the effectiveness of error-correction we performed the 'unsupervised' (U) experiment, where the tracker is initialized only once at the beginning. In addition, to demonstrate the real time effectiveness of our tracker we also performed the 'realtime' (R) experiment on the VOT 2017 dataset.

To demonstrate the advantages of each modification we start with a baseline tracker that uses MOSSE with fHoG features along with PCA as implemented in fDSST but without the scale filter. The tracker thus obtained we call MOSSE++^{i} where superscript 'i' indicates the padding size employed in the tracker. In this experiment we have studied the effect of padding 1 (used in DSST) and 1.5 (used in KCF). We have added the scale estimation to both the trackers described above as MOSSE++Scale^i. The proposed tracker combines both MOSSE++Scale^1 and MOSSE++Scale$^{1.5}$. A comparison of these is shown in Table 1.

Table 1. Performance comparison with trackers having different padding size with and without scale estimation on VOT 2016.

Tracker	EAO (U)	EAO (B)
MOSSE++Scale^1	0.419	0.168
MOSSE++^{1}	0.412	0.159
MOSSE++Scale$^{1.5}$	0.364	0.157
MOSSE++$^{1.5}$	0.333	0.150
Proposed	**0.454**	**0.195**

These results show that the EAO improves from 0.159 to 0.195 for the proposed tracker, when compared to the baseline tracker MOSSE++^{1}. Improvements due to the scale addition can be clearly seen as it has improved EAO from 0.333 (MOSSE++$^{1.5}$) to

0.364 (MOSSE++Scale$^{1.5}$). Effect of padding size can also be observed (keeping scale fixed to a constant size). When padding is reduced from 1.5 to 1 the EAO increased from 0.333 (MOSSE++$^{1.5}$) to 0.412 (MOSSE++^{1}). One probable explanation is the leaking of background noise into the higher padding filter. The proposed tracker which combines both the filters along with the scale has the highest EAO 0.454. This effect can be seen in the Fig.4, where MOSSE++Scale^{1}, shown in Fig.4.a,b,c., has failed; MOSSE++Scale$^{1.5}$, shown in Fig.4.d,e,f, has also failed. However the proposed tracker succeeds, as shown in Fig.4.g,h,i.

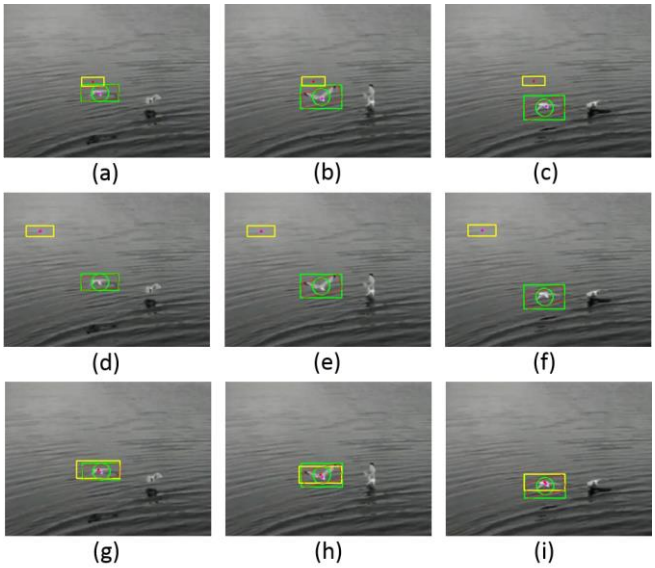

Fig. 4: Resilience of tracker using two filters (g,h,i) compared with trackers using a single filter (a,b,c and d,e,f). Green box is ground truth and yellow corresponds to tracker estimate.

A second set of experiments demonstrate the advantages derived from our combined approach in comparison with VOT 2016 results of DSST2014 and KCF2014, along with the generated results of the publicly released fDSST code*. KCF2014 is an upgraded version of KCF with multi-scale support and sub-cell peak estimation [7]. Table 2 shows the performance measures achieved for the experiment, together with performance of the existing state-of-the-art techniques. As seen in this table, our approach achieves higher EAO and EFO than DSST, fDSST and even KCF2014.

Table 2. Performance comparison with other correlation based trackers submitted in VOT 2016 [7]

Tracker	R	A	EAO (B)	EFO
KCF2014	1.95	0.48	0.192	21.79
SAMF2014	1.91	0.50	0.186	4.01
DSST2014	2.38	0.52	0.181	12.75
ART_DSST	2.51	0.50	0.167	8.45
sKCF	2.86	0.48	0.153	91.06
fDSST*	2.64	0.49	0.164	13.02
Proposed	**2.03**	**0.48**	**0.195**	**42.51**

The third set of experiments demonstrate the performance improvement obtained on the VOT 2017 benchmark in comparison with DSST and fDSST only, as the other trackers from VOT 2016 have not been evaluated in VOT 2017 [38]. As can be seen in Table 3 the proposed tracker achieves higher EAO on all three experiments (baseline, realtime and unsupervised) of the VOT 2017. These clearly demonstrate the improved error correction and speed of the tracker when compared with DSST and fDSST.

Table 3. Performance comparison with other correlation based trackers on VOT 2017 [38]

Tracker	EAO (B)	EAO (R)	EAO (U)
DSST	0.079	0.077	0.256
fDSST*	0.099	0.088	0.346
Proposed	**0.116**	**0.113**	**0.399**

Finally, Table 4 shows that on VOT 2017, the proposed tracker significantly outperforms the top trackers (from the baseline challenge) on the realtime challenge because those trackers are slow due to the use of deep neural network features or multilevel features [38][33]. Evidently these are highly suitable for offline tracking but their performance drops drastically for real time tracking leading to the proposed tracker having better EAO in real time.

Table 4. Performance comparison with the top performers submitted in VOT 2017 [38]

Tracker	EAO (Baseline)	EAO (Realtime)
LSART	0.323	0.055
CFWCR	0.303	0.062
CFCF	0.286	0.059
ECO	0.280	0.078
Gnet	0.274	0.060
MCCT	0.270	0.060
CCOT	0.267	0.058
Proposed	**0.116**	**0.113**

Hence the highlight of our approach is the improvement in speed, ~3x with respect to DSST and the error correction method making our algorithm more robust and suitable for real time tracking.

4. CONCLUSION

In this paper we have proposed a new methodology for tracking using multiple correlation filters which enables us to not only track objects in real time but also correct for errors made in the past by combining the information from multiple trackers in the light of newer frames. The traditional scale handling approach has also been sped up. This methodology has been evaluated on the VOT 2016 and 2017 benchmarks and it has been found that significant improvements are obtained in speed along with gains in performance in challenging sequences involving partial occlusions, deformations and out of plane rotations. In future we plan to investigate the applicability of the proposed algorithm in other methodologies like Staple and FSRDCF. We also plan to integrate object in-plane rotation handling.

*fDSST code for this experiment was obtained from www.cvl.isy.liu.se/en/research/objrec/visualtracking/scavistrack/fDSST_code.zip

5. REFERENCES

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

[2] D. S. Bolme, B. A. Draper, and J. R. Beveridge, "Average of synthetic exact filters," *CVPR*, 2009.

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

[4] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. "High speed tracking with kernelized correlation filters". *TPAMI*, 37(3):583–596, 2015.

[5] Y. Li and J. Zhu. "A scale adaptive kernel correlation filter tracker with feature integration". *ECCV Workshop*, 2014.

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

[7] M. Kristan, A. Leonardis, J. Matas, R. Felsberg, Pflugfelder, M., L. Cehovin, Vojir T.and G. Hager, and et al. "The visual object tracking vot2016 challenge results". *ECCV workshop*, 2016.

[8] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Nebehay, R. Pflugfelder, and G. Hager. "The visual object tracking vot2015 challenge results". *ICCV workshop*, 2015.

[9] Y. Wu, J. Lim, and M.-H. Yang. "Online object tracking: A benchmark". *CVPR*, 2013.

[10] Y. Wu, J. Lim, and M.-H. Yang. "Object tracking benchmark". *TPAMI*, 37(9):1834–1848, 2015.

[11] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. "Discriminative scale space tracking". *TPAMI*, pp(99), 2016

[12] S. Hare, A. Saffari, and P. Torr. "Struck: Structured output tracking with kernels". *ICCV*, 2011

[13] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". *CVPR*, 2005

[14] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. "Fully-convolutional siamese networks for object tracking". *ECCV workshop*, 2016.

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

[16] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[17] H. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," *ICCV*, 2013.

[18] J. Henriques, J. Carreira, R. Caseiro, and J. Batista, "Beyond hard negative mining: Efficient detector learning via block-circulant decomposition," *ICCV*, 2013.

[19] B. Babenko, M.-H. Yang, and S. Belongie. "Visual Tracking with Online Multiple Instance Learning". *CVPR*, 2009

[20] Wu, Y., Lim, J., Yang, M.H.: "Online object tracking: A benchmark". *CVPR*. (2013)

[21] Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Cehovin, L., Nebehay, G.,Vojir, T., G., F., et al.: "The visual object tracking vot2014 challenge results". *ECCV2014 Workshops, Workshop on visual object tracking challenge*. (2014)

[22] Danelljan, M., Robinson, A., Shahbaz Khan, F., Felsberg, M.: "Beyond correlation filters: Learning continuous convolution operators for visual tracking". *ECCV*.(2016)

[23] Nam, H., Baek, M., Han, B.: "Modeling and propagating cnns in a tree structurefor visual tracking". *CoRR* abs/1608.07242 (2016)

[24] Nam, H., Han, B.: "Learning multi-domain convolutional neural networks for visual tracking". *CoRR*. (2015)

[25] Van de Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: "Learning color names for real-world applications". *IEEE Transactions on Image Processing* 18(7) (2009)1512–1524

[26] Wang, L., Ouyang, W., Wang, X., Lu, H.: "Visual tracking with fully convolutional networks". *ICCV*. (2015)

[27] Wang, L., Ouyang, W., Wang, X., Lu, H.: "Stct: Sequentially training convolutional networks for visual tracking". *CVPR*. (2016)

[28] Zhizhen Chi, Hongyang Li, Huchuan, and Ming-Hsuan Yang. "Dual deep network for visual tracking", *IEEE Trans. on Image Processing,* 2017.

[29] Lee, J.Y., Yu, W.: "Visual tracking by partition-based histogram back projection and maximum support criteria". *Proceedings of the IEEE International Conference on Robotics and Biomimetic (ROBIO)*. (2011)

[30] Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: "Learning spatially regularized correlation filters for visual tracking". *ICCV* (2015)

[31] A. Lukežič, T. Vojīr, L. Čehovin, J. Matas, M. Kristan, "Discriminative correlation filter with channel and spatial reliability", arXiv, 2016.

[32] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". *ICLR*, 2015.

[33] HK Galoogahi, A Fagg, C Huang, D Ramanan. "Need for Speed: A Benchmark for Higher Frame Rate Object Tracking", arXiv, 2017.

[34] Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Cehovin, L.: "A novel performance evaluation methodology for single-target trackers." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2016

[35] Cehovin, L., Leonardis, A., Kristan, M.: "Visual object tracking performance measures revisited." *IEEE Transactions on Image Processing,* 25(3), 2015

[36] Cehovin, L., Kristan, M., Leonardis, A.: "Is my new tracker really better than yours?" *WACV 2014: IEEE Winter Conference on Applications of Computer Vision,* 2014

[37] Montero, A.S., Lang, J., Laganiere, R.: "Scalable kernel correlation filter with sparse feature integration". *The IEEE International Conference on Computer Vision (ICCV) Workshops*.

[38] Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder et al.: "The visual object tracking vot2017 challenge results". *ICCV2017 Workshop (2017*).