Appearance Changes Detection during Tracking

Wei Chen Xifeng Guo Xinwang Liu En Zhu Jianping Yin
College of Computer
National University of Defence Technology
Changsha, China
Email: cwintardis@gmail.com

***Abstract*—Correlation tracker has made a huge success in visual object tracking. However, it is mainly because that the tracker cannot catch the occurrence of appearance changes, tracking based on correlation filters often drifts due to the unexpected appearance changes caused by occlusion, deformation and background clutter. In this paper, we propose a new method to detect the case when the tracker encountered the unexpected appearance changes. This method uses the following points:1)Filter response curve would decreases dramatically when target suffers heavy appearance changes. 2)Features extracted from deeper layers of convolutional neural networks (CNNs) have more semantics information and features extracted from shadower layers have more spatial information. Extensive experimental results on several public benchmark datasets show that the proposed method can deal with the appearance changes effectively.**

I. Introduction

Visual tracking is an essential component of numerous practical applications, such as video surveillance, driverless car and human-computer interaction [1], and so on. Even though much progress has been made in recent years, it is still a difficult problem to design a good tracking algorithm for all scenarios. The main challenge is that the algorithms cannot well deal with severe appearance changes.

Recently, due to the huge success of deep neural networks in the field of image [2] and voice processing, the features based on CNNs have been used into object tracking field to deal with appearance changes of target, and have obtained state-of-the-art results [3], [4], [5], [6]. It is thus of great value to know how to better use the features extracted from CNNs.

From the existing literature [3], [4] we know that features extracted from CNNs have the property [3] that the features in deeper layers have more semantic information and features in earlier layers have more spatial information. This property can be used to track in correlation filters tracking framework.

Real time performance is important for algorithms in many fields [7], [8]. For the high real time performance of the tracker based on correlation filters [9], [8], it has been used as base tracker in many tracking algorithms [10], [11], [3], which have gain state-of-arts results. It is naturally to combine the CNNs features and correlation tracker together to achieve better tracking results [3].

However, some issues ensue with above methods. To begin with, tracker based on correlation have no mechanism to detect occlusion and other appearance changes. When the tracker encounters a long time occlusion, tracker may drift to inappropriate positions. The second issue is that the trackers with CNNs features does not fully use the property of the CNNs features, which would not play the full part of the features.

In this paper, we alleviate these two issues by the proposed methods: 1) Use the properties of the correlation filter response curve to detect the occlusion, instead of just finding the highest response in each frame. 2) Use the property of the features from CNNs to detect the deformation of the target.

The main contributions of this paper are as follows. First, we find a new property of the correlation filter response curve which can be used to detect the occlusion with no extra computation. This is important to the real performance of the tracking algorithm. Second, we use the property of the features from CNNs to detect the deformation of the target which exploits the property further and provides a new clue to detect appearance changes. Third, extensive experiments on benchmark datasets [12] show that the proposed methods are effective.

II. Related Work

In this section, we briefly review KCF [8] and HCFT [3] methods, which are the baselines of our methods.

A. KCF method

KCF tracker is the kernel version of CSK tracker [13], which is a typical correlation tracker. Many tracking algorithms are based on this algorithm [10], [11], [3], [14]. In CSK tracker, the training samples are generated by considering all the circular shifts of target zone $\mathbf{x}$ (this zone includes the target and bigger than the target)along the M and N dimensions, where M and N are the width and height of x, respectively. For each sample $\mathbf{x}_{m,n}$,$(m,n) \in \{0,1,...,M-1\} \times \{0,1,...,N-1\}$, there has a Gaussian function label $y(m,n)$,which is computed by:

$$y_{m,n} = exp(-\frac{(m-M/2)^2 + (n-N/2)^2}{2\sigma^2}) \quad (1)$$

where σ is the kernel width. Then, the tracking task can be converted into a regression problem. The CSK algorithm learns a discriminative classifier $\mathbf{w}$ by:

$$\min_{\mathbf{w}} \sum_{m=1,n=1}^{M,N} \|\mathbf{w} * \mathbf{x}_{m,n} - y_{m,n}\|^2 + \lambda \|\mathbf{w}\|^2 \quad (2)$$

where λ is a regularization parameter. In [13] this problem is solved by using Fourier transformation(FFT). Here we use capital letters to denote the discrete Fourier transform (DFT) of a vector. In frequency domain the classifier $\mathbf{w}$ can be written as:

$$\mathbf{W} = \frac{\mathbf{Y} \odot \mathbf{X}^*}{\mathbf{X} \odot \mathbf{X}^* + \lambda} \tag{3}$$

where Y indicates discrete Fourier transform (DFT) of label $\mathbf{y} = \{y_{m,n} | (m,n) \in \{0,1,...,M-1\} \times \{0,1,...,N-1\}\}$. $\mathbf{X}$ denotes the DFT of target zone $\mathbf{x}$, and $\mathbf{X}^*$ denotes the complex-conjugate of $\mathbf{X}$. When given an image patch (the patch includes the target zone and bigger than the target zone) $\mathbf{z}$ in next frame, the correlation response map R can be obtained by:

$$R = \mathcal{F}^{-1}(\mathbf{W} \odot \mathbf{Z}) \tag{4}$$

where $\mathcal{F}^{-1}$ denotes the inverse FFT transform and $\mathbf{Z}$ denotes the DFT of $\mathbf{z}$. Then, CSK tracker estimates the position of target by searching the maximum value in response map R.

For response map R, the KCF tracker only finds its maximum value in each frame, while does not use other properties that we used in our method to detect the occlusion.

B. HCFT method

The HCFT tracker uses the KCF method as the base tracker and use the feature extract from the CNNs instead of the HOG features to describe the target. HCFT is based on the observation that the features from the last layers of CNNs encode semantic abstraction of targets and are robust to appearance variations. On the other hand, the features from early layers retain more fine-grained spatial details and thus are useful for precise localization [3]. HCFT tracker combines the response of each layer by:

$$\max_{m,n} f_{l-1}(m,n) + \gamma f_l(m,n) \tag{5}$$

where $f_l(m,n)$ denotes the location of the maximum values of response R on l-th layer and γ is a regularization term. The main steps of HCFT algorithm can be shown as Fig. 1 [3].

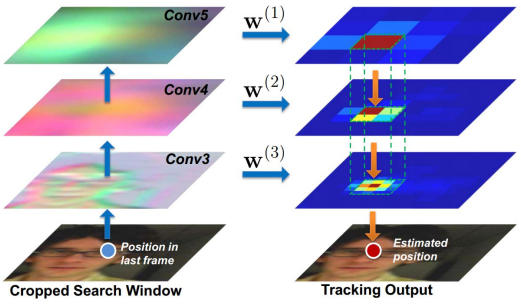


Fig. 1. Main steps of HCFT algorithm. First, crop search window according to the target position in the previous frame. Second, use several convolutional layers(here are third, fourth and fifth layers) as the representations of the target. Then each layer i convolved with the filter $w_{(i)}$ to generate a response map. Third, use Eq. 5 to calculate the target position in current frame

Based on the observation in [3] we found that the similarity between different features extract from different layers has closed relationship with appearance changes of the target. We use this similarity to detect deformation of target during tracking process.

III. Proposed Method

For each candidate i in frame t, $\mathbf{W}$ in Eq. 3 can be written as $\mathbf{W}_{ti} = [w_a, w_b, ..., w_{M_w * N_w}]$, Z can be written as $\mathbf{Z}_{ti} = [z_a, z_b, ..., z_{M_z * N_z}]$, where $[w_a, w_b, ..., w_{M_w * N_w}]$ and $[z_a, z_b, ..., z_{M_z * N_z}]$ denote the elements in $\mathbf{W}_{ti}$ and $\mathbf{Z}_{ti}$, respectively. Then, for each candidate i according to the property of the FFT [15], the response of candidate i can be written as:

$$\begin{aligned} R_{ti} =& \mathcal{F}^{-1}(\mathbf{W}_{ti} \odot \mathbf{Z}_{ti}) \\ =& \mathcal{F}^{-1}(\mathbf{W}_{ti} \odot \mathbf{Z}_{ti \neq 0}) + \mathcal{F}^{-1}(\mathbf{W}_{ti} \odot \mathbf{Z}_{ti=0}) \\ =& \mathcal{F}^{-1}(\mathbf{W}_{ti} \odot \mathbf{Z}_{ti \neq 0}) \end{aligned} \tag{6}$$

where $\mathbf{Z}_{ti \neq 0}$ means that the elements in $\mathbf{Z}_{ti}$ belong to the target. $\mathbf{Z}_{ti=0}$ means that the elements in $\mathbf{Z}_{ti}$ do not belong to the target. The response map R_t of each frame t can be obtain by:

$$\mathbf{R}_t = [R_{t1}, R_{t2}, R_{t3}, ..., R_{ti}, ..., R_{tM*N}] \tag{7}$$

where $M * N$ is the number of the candidate. Here we assume that the maximum value in $\mathbf{R}_t$ is R_{ti}. When the target suffers some severe appearance changes, the number of zero elements in $\mathbf{Z}_{ti}$ would increase dramatically which would lead to a dramatic decrease of the value of R_{ti}. Then, the response curve would have a dramatic decrease in frame t.

In order to better understand this property of response curve, we show it in Fig. 2. We obtain the response curve in Fig. 2 by calculating the response of ground truth in each frame with the ground truth in first frame. The horizontal axis is the frame number and the vertical axis is the maximum value of response in each frame. The red boxes are the mutation regions. Region 2,4,7 and 10 are the normal conditions. Region 1 and 3 are the heavy deformation of the target. Region 5 and 6 are in-plane rotation of the target. Region 8 and 9 are the occlusion conditions.

From Fig. 2, we observe that when the target suffers some appearance changes such as occlusion, in-plane rotation, deformation and other unexpected reasons, the response curve would have a dramatic decrease. Furthermore, we observe that when the target recovers from abnormal conditions, the response curve would have a dramatic increase. Thus, we could use this property to deal the abnormal conditions such as occlusion of the target to help the tracker to improve robustness.

A. Occlusion detection mechanism

When heavy occlusion happen, most of the target zone would be the background. Here we assume $\mathbf{Z}_{ti}$ is the target zone. In such case, the number of $\mathbf{Z}_{ti=0}$ in $\mathbf{Z}_{ti}$ would have a sudden increase that would lead to the dramatic decrease of the response curve of $\mathbf{Z}_{ti}$. Take Fig. 2 for example, when the man who wears a glass blocked the girl who is the foreground in

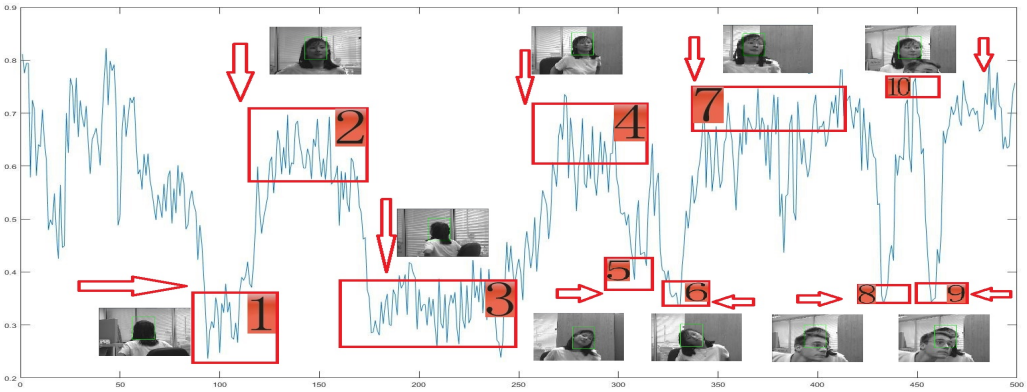


Fig. 2. The response curve of *Girl* sequence

image sequence *Girl*, the curve of the response has a dramatic decrease in Region 8 and 9. When this mutation point occur, we use the Eq. 8 to determine what the tracker should do:

$$\begin{cases} flag = 1, upda = 1 & \epsilon 1 < Res_t \\ flag = 1, upda = 0 & \epsilon 2 < Res_t \leq \epsilon 1 \\ flag = 0, upda = 0 & Res_t < \epsilon 2 \end{cases} \tag{8}$$

where the value of $flag$ denotes whether the tracker keep tracking, and the value of $upda$ denotes whether the tracker update its tracking model. Here "1" means true, "0" means false. $\epsilon 1$ is the threshold that judges when occlusion happens and $\epsilon 2$ is the threshold that judges when the complete occlusion happens. When complete occlusion happen, besides stop tracking and updating, we also expand the search area of the algorithm to prevent the target reappearing in another position.

B. Deformation detection mechanism

Deformation of the target is another challenge in tracking filed. How to deal with the deformation is very important to a tracker.

Features from CNNs has some special properties shown in Fig. 3 [3]. The features from the deeper layers have

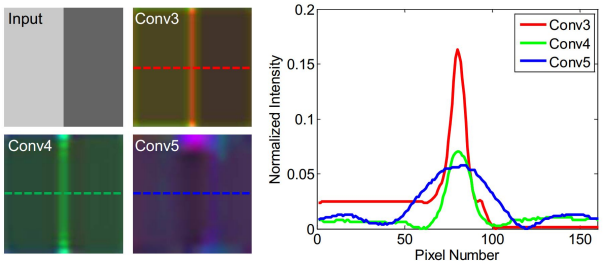


Fig. 3. The property of CNNs features

more semantics information and they are less sensitive to the deformation. But the features from the shadower layers are more sensitive to the deformation. This property can be used to detect the deformation of the target by:

$$De_t = corr2(feature_s, feature_d) \tag{9}$$

where De_t is the degree of the deformation. Here we use the similarity between $feature_s$ and $feature_d$ to measure the degree of deformation, where $feature_s$ and $feature_d$ denote the features from shadow layers and deep layers, respectively. Then, we use Eq. 10 to control the update scheme of the tracking algorithm.

$$\begin{cases} \text{Updating} & De_t > \varepsilon, \\ \text{Stoping updating} & others \end{cases} \tag{10}$$

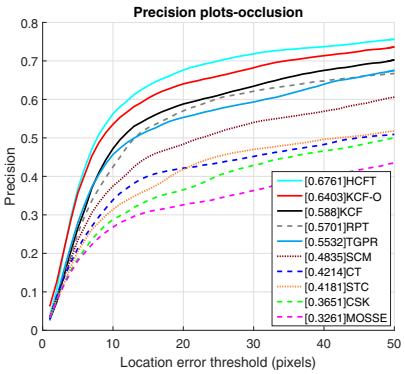
where ε is a threshold. When the De_t is less than ε, the target suffers a violent deformation. In such case the algorithm should stop updating the target model to prevent bringing some wrong information to the target model.

IV. Experiments

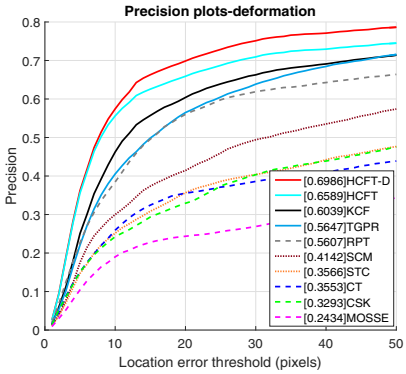
The proposed methods are implemented in MATLAB and run on a PC with a 3.4GHz CPU and 16GB RAM. The parameters mentioned above are set as follow:$\epsilon 1$ is 0.3 and $\epsilon 2$ is 0.2. In Eq. 10 ε is 0.9. These three value are empirical value.

We evaluate the proposed occlusion detection method on 29 image sequences which are annotated with the occlusion attribute in the benchmark data set [12][1] and on the all data set containing 50 image sequences. Then evaluate the proposed deformation detection method on 23 image sequences which are annotated with the deformation attribute in the benchmark data set [12] and on the all data set containing 50 image sequences. For comparison, we also run nine state-of-the-art algorithms on that data set. These algorithms are HCFT [3], KCF [8], CT [10], RPT [16], CSK [13], MOSSE [13], SCM [17], TGPR [18], and STC [19] methods.

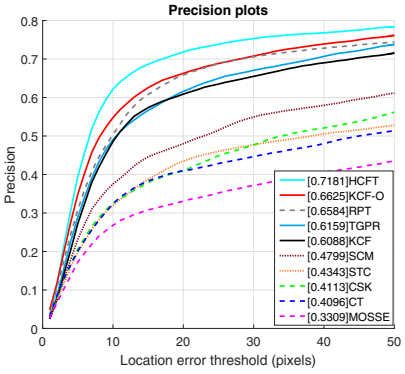
[1]http://www.visual-tracking.net



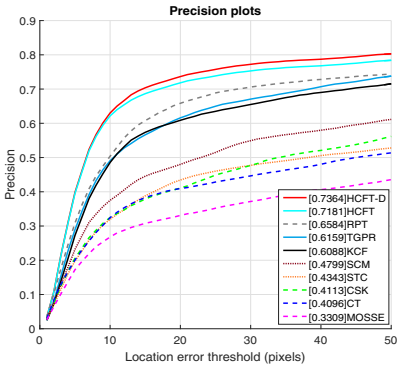
(a) precision plots over 31 occlusion sequences of 50 benchmark sequences [12]



(b) precision plots over 23 deformation sequences of 50 benchmark sequences [12]



(c) precision plots over 50 benchmark sequences [12]



(d) precision plots over 50 benchmark sequences [12]

Fig. 4. Distance precision plots

Quantitative Evaluation: We evaluate the above-mentioned algorithms and our methods using the center location error, and the results are shown in Fig. 4.

From Fig. 4(a,c), we know that when use our occlusion detection mechanism (KCF-O), the precision rate of KCF algorithm improved obviously and the proposed method can still valid in all 50 sequences . And in Fig. 4(b,d), when add our deformation detection mechanism to HCFT algorithm (HCFT-D), the precision rate of this algorithm has been improve obviously and keep its advantage in all 50 sequences.

Qualitative Evaluation: We add our mechanism to the base trackers and then compare the tracking results of them on several challenging sequences with base tracker KCF [8] and HCFT [3] .The KCF algorithm is based on a correlation filter and uses HOG features to represent the candidate. Thus it is a robust tracker in many scenarios but it can not handle the heavy occlusions very well. As shown in the images sequences in Fig. 5(a), when the target recovery from the heavy occlusion it drifts. However, in Fig. 5(a), after adding our occlusion detection mechanism, the KCF algorithm can tell when the tracker suffers occlusion (frame 433) and can still find the right position of target after the occlusion(frame 450) (*Girl* sequence). In *Liquor* sequence of Fig. 5(a), when occlusion happen, KCF-O can still track the target but KCF failed to track. HCFT method is a tracker which uses deep learning features as the target representation, its performance is even better than the improved method KCF-O as show in Fig. 4(a,c). However, when suffered heavy deformation it could drift as shown in Fig. 5(b). But after adding our deformation detection mechanism as shown in Fig. 5(b), HCFT-D could still track the target after occlusion and deformation, overcoming the failure cases mentioned in [3].

V. Conclusions

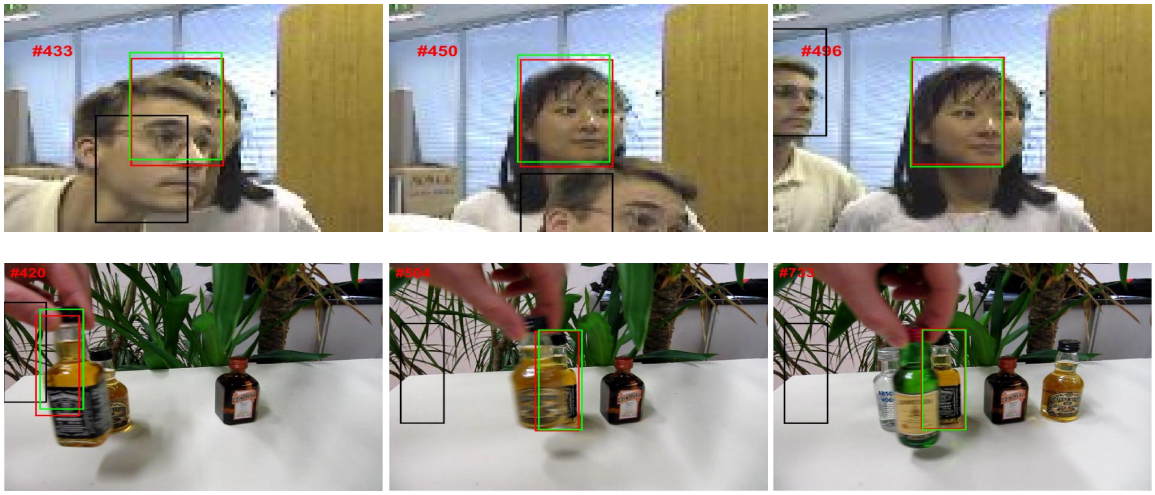
In this paper, we propose two mechanisms to detect occlusion and deformation of target. We exploit the property of correlation filter response curve and the features extracted from CNNs. Extensive experiments have shown that our methods can well deal with the occlusion and deformation. Since our mechanisms are independent, they can be used in any tracking method lacking this component.

Acknowledgment

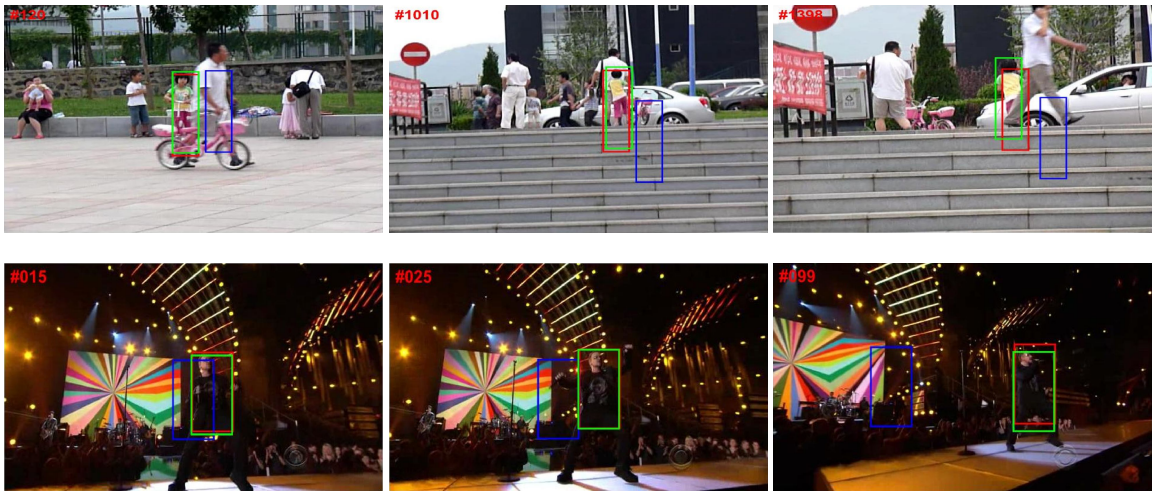
This work was supported by the National Natural Science Foundation of China (Project no. 61170287, 61232016).

References

[1] X. Li, Y. Dou, X. Niu, J. Xu, and R. Xiao, "An efficient robust eye localization by learning the convolution distribution using eye template," *Computational intelligence and neuroscience*, vol. 2015, 2015.



(a) Qualitative evaluation of occlusion detection mechanism. Red boxes show our results, green ones are ground truth and black ones are the original tracker (KCF)



(b) Qualitative evaluation of deformation detection mechanism.Red boxes show our results, green ones are ground truth and blue ones are the original tracker (HCFT)

Fig. 5. Qualitative evaluation on several challenging sequences

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3074–3082.

[4] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3119–3127.

[5] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," *Advances in Neural Information Processing Systems*, pp. 809–817, 2013.

[6] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1424–1435, 2015.

[7] R. Hu, G. Liu, J. Jiang, and L. Wang, "G2lc: Resources autoscaling for real time bioinformatics applications in iaas," *Computational and mathematical methods in medicine*, vol. 2015, p. 1, 2015.

[8] J. F. Henriques, R. Caseiro, P. Martins, and Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*. Springer, 2012, pp. 702–715.

[10] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1090–1097.

[11] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *International Conference on Computer Vision*, 2015.

[12] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

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

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

[15] A. V. Oppenheim, R. W. Schafer, J. R. Buck *et al.*, *Discrete-time signal processing*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 2.

[16] Y. Li, J. Zhu, and S. C. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 353–361.

[17] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1838–1845.

[18] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *European Conference on Computer Vision*. Springer, 2014, pp. 188–203.

[19] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *European Conference on Computer Vision*. Springer, 2014, pp. 127–141.

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