Scale-Invarinat Kernelized Correlation Filter using Convolutional Feature for Object Tracking

Mingjie Liu, Cheng-Bin Jin, Bin Yang, Xuenan Cui and Hakil Kim
Information and Communication Engineering, Inha University, 100 Inha ro, Namgu 22212, Incheon, Republic of Korea

Keywords: Object Tracking, Kernelized Correlation Filter, Convolutional Features, Scale Variation, Appearance Model Update Strategy.

Abstract: Considering the recent achievements of CNN, in this study, we present a CNN-based kernelized correlation filter (KCF) online visual object tracking algorithm. Specifically, first, we incorporate the convolutional layers of CNN into the KCF to integrate features from different convolutional layers into the multiple channel. Then the KCF is used to predict the location of the object based on these features from CNN. Additionally, it is worthying noting that the linear motion model is applied when do object location to reject the fast motion of object. Subsequently, the scale adaptive method is carried out to overcome the problem of the fixed template size of traditional KCF tracker. Finally, a new tracking update model is investigated to alleviate the influence of object occlusion. The extensive evaluation of the proposed method has been conducted over OTB-100 datasets, and the results demonstrate that the proposed method achieves a highly satisfactory performance.

1 INTRODUCTION

Visual object tracking, which is a crucial component of computer vision system, has widespread applications, including surveillance, traffic control, and automatic drive (Yilmza, 2006), (David, 2008), (Arnold, 2014). The adoption of the online-based discriminative learning method has arguably constituted a breakthrough in visual object tracking (Babenko, 2009), (Hare, 2016), (Zhang, 2014). Given an initial image patch containing the target, the classifier is trained to distinguish the appearance and its background. This classifier can be evaluated exhaustively in many locations, in order for detection to be carried out in subsequent frames. After locating the object's position, a new image patch is provided to update the model.

Although the discriminative learning method has resulted in a great deal of progress in visual object tracking, issues of tracking bounding box drift and object loss still occur as a result of factors such as occlusion, scale variance, background clutter, and illumination changes. Specifically, the root causes of these issues from a features view can mainly be attributed to the following: (i) object features are not robust enough to distinguish the object from the background; (ii) ideal features cannot be obtained because of the occlusion; (iii) object appearance is changed drastically by pose, illumination, and scale, among others. In general, the features extracted from training data cannot adequately describe the object, which results in the object imprecisely being located from a given image patch.

To solve those problems, different machine learning-based algorithms are proposed. Hare et al. (Hare, 2016) present a framework for adaptive visual object tracking using structured output prediction, which is based on a kernelized structured output SVM to provide adaptive tracking. Henriques et al. (Henriques, 2015) take advantage of the fact that the convolution between two image patches in time domain can be transformed into an element-wise product in frequency domain, which can specify the desired output of a linear classifier for several translations, or image shifts, simultaneously. To solve occlusion problem, Yang et al. (Yang, 2016) recently extend the correlation filter-based method through fusing the feature color name histograms of oriented gradients (CN-HOG), which consists of CNs and HOG, and is robust to partial occlusion.

Liu, M., Jin, C., Yang, B., Cui, X. and Kim, H.
Scale-Invarinat Kernelized Correlation Filter using Convolutional Feature for Object Tracking.
DOI: 10.5220/0006694003350340
In *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018)*, pages 335-340
ISBN: 978-989-758-293-6

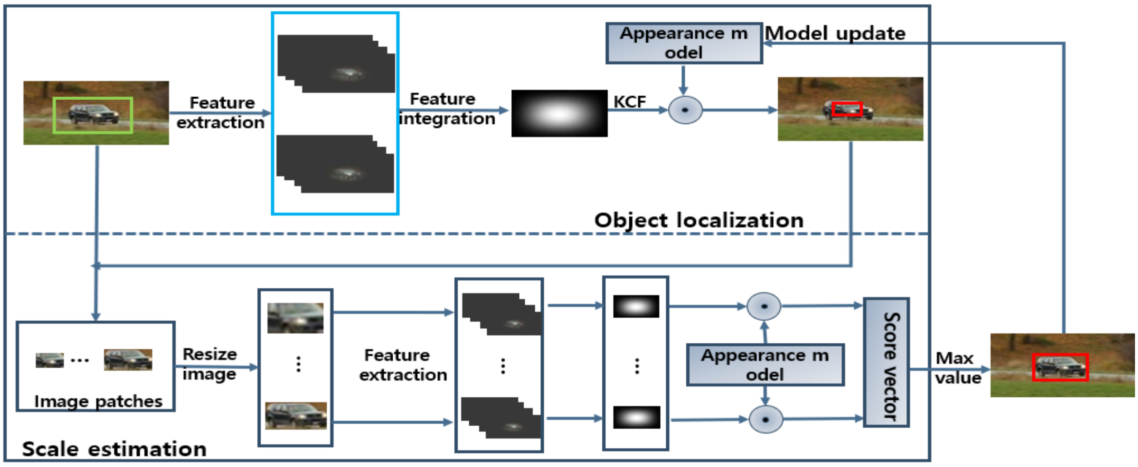


Figure 1: Object tracking system overview.

Correlation filter based tracking algorithm have shown favorable performance recently. Nonetheless, scale variance and tracking model update strategy limit their accuracy enhancement. In this paper, we separate tracking algorithm into two parts: object localization and scale estimation. During object localization, motion model is applied to locate the search window, which can solve fast motion problem. Since multiple channels can be applied in kernelized correlation filter (KCF) for the image features, features extracted from the different convolutional layers in search window are integrated into the object representation to improve the tracking accuracy, and it is then convolved with the KCF tracker to generate a response map, in which the maximum value of location indicates the estimated target position. Then, a scale adaptive method is proposed to overcome the problem of the fixed template size in the traditional KCF tracker. Furthermore, a new tracking model update strategy is exploited to avoid the model corruption problem.

2 PROPOSED METHOD

With the aim of locating the object precisely and solving the scale variation problem, we separate tracking algorithm into two parts: object localization and scale estimation. Figure 1 illustrates the main steps of our method.

2.1 Object Localization

During object localization, the search window is first estimated by motion model to overcome the fast motion problem. Since multiple channels can be applied in KCF for the image features, features extracted from the different convolutional layers in search window are integrated into the object representation, and it is then convolved with the KCF tracker to generate a response map, in which the maximum value of location indicates the estimated target position.

2.1.1 Motion Model

The location of the object is predicted in a search window which is usually obtained by padding object position got from previous frame. However, if the object moves fast, it will be out of the search window. To this point, a simple linear motion model with constant velocity is applied to roughly estimate the location of object in frame t based on the predicted position in frame $t-1$, and then search window is obtained by padding predicted object position based on motion model.

Given the velocity v_{t-1} in frame $t-1$, the velocity v_t in frame t is updated as

$$\begin{aligned} \hat{v}_t &= T(p_t - p_{t-1}) \\ v_t &= \alpha v_{t-1} + (1-\alpha)\hat{v}_t \end{aligned} \tag{1}$$

where $p_t = [x_t, y_t]$ is the center location of the object at frame t and $T = \frac{1}{F}$ is the time gap between two adjacent frames, F is frame rate of the video.

After getting the velocity of the object in frame t, the predicted location of the object by motion model at frame $t+1$ is defined as

$$p_{t+1} = p_t + v_t T \tag{2}$$

2.1.2 Kernelized Correlation Filter

Viewing the correlation filters as classifiers, they can be trained by finding the relation between the i-th input $\mathbf{x}_i$ and its regression target y_i from the training set, and then use linear regression to finish prediction (Li, 2014). To utilize correlation filter to linear non-separable samples, we hope to find a nonlinear mapping function (kernel) which can map those samples to a higher dimension to make them linear separable, which is called kernelized correlation filter (KCF). The key innovation of KCF is the use of the structure of circulant matrices. A circulant matrix is used to learn all the possible shifts of the object from base sample. The coefficient $\hat{\alpha}$ encodes the training samples, consisting of all shifts of base sample in the frequency domain. The learning equation is expressed as

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathrm{k}}^{\mathrm{xx}'} + \lambda} \tag{3}$$

where ^ denotes the Discrete Fourier Transform (DFT) of a vector. $\mathrm{k}^{\mathrm{xx}'}$ is the kernel correlation function between signals $\mathbf{x}$ and $\mathbf{x}'$. The training label matrix $\mathbf{y}$ follows a Gaussian distribution that smoothly decays from the value of 1 for the estimated target bounding box to 0 for other shifts. The division represent an element-wise division and λ is a regularization parameter that controls overfitting.

The final object position is determined by $\delta = \arg\max(f(\hat{\mathrm{k}}^{\mathrm{xz}}))$ between the current and the next patch. z is the spatial index with maximum response in $f(\hat{\mathrm{k}}^{\mathrm{xz}})$. The response for each location is

$$f(\hat{\mathbf{k}}^{\mathrm{xz}}) = F^{-1}(\hat{\mathbf{k}}^{\mathrm{xz}} \odot \hat{\boldsymbol{\alpha}}). \tag{4}$$

2.1.3 Convolutional Feature Integration

Based on the deep feature analysis for object tracking in (Wang, 2015), we know that different layers encode different feature types: deeper layers capture the semantic concepts of object categories, while lower layers encode more discriminative features. Convolutional layers from VGG (Simonyan, 2014) are applied to represent object appearance. We focus only on the accurate object position in the tracking. In this case, fully connected layers can be removed to save processing time, as they exhibit little spatial resolution.

The outputs of the selected convolutional layers are used as multi-channel features. Suppose the multiple data representation channels are concatenated into a vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D]$, the Gaussian kernel correlation function $\mathbf{k}^{\mathrm{xx}'}$ is defined as

$$\mathbf{k}^{\mathrm{xx}'} = \exp\left(-\frac{1}{\sigma^2}(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2F(\sum_D (\hat{\mathbf{x}}^D)^* \odot (\mathbf{x}'^D))\right) \tag{5}$$

The features from different convolutional layers are complementary to another; therefore, our method can fuse both the semantic and discriminative information for object location. It should be noted that spatial resolution is gradually reduced with an increase in convolutional layer depth due to the pooling operators used in CNN, which results in insufficiently accurate object location. This issue is addressed by resizing each feature map to a fixed, larger size. Let $\mathbf{h}$ denote the feature map and $\mathbf{x}$ be the upsampled feature map; then, the feature vector for the i-th location is

$$\mathbf{x}_i = \sum_k \beta_{ik} \mathbf{h}_k \tag{6}$$

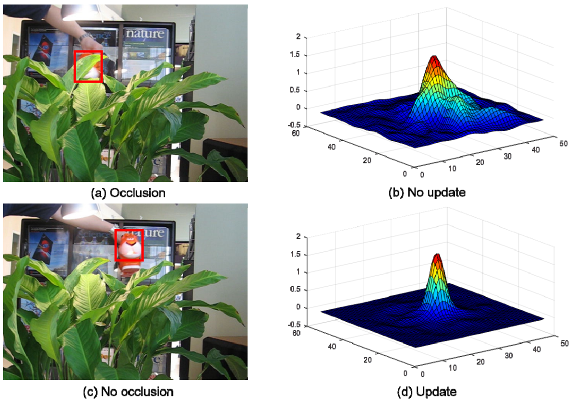
where the interpolation weight β_{ik} depends on the positions of the neighboring feature vectors i and k, respectively. Note that this interpolation occurs spatially, and can be seen as location interpolation.

2.2 Scale Estimation

Scale variation poses a significant challenge in correlation filter-based trackers. The desired tracker should not only precisely locate the object, but also be adaptive to the object size variations.

Following object location, we employ object padding to enlarge the image representation space. Let $\mathrm{s}_\mathrm{T} = (s_x, s_y)$ denote the current frame's template size and N be the number of scales. For each $n \in \left\{\left\lfloor -\frac{N-1}{2} \right\rfloor, \ldots, \left\lfloor \frac{N-1}{2} \right\rfloor\right\}$, an image patch J_n of size $a^n s_x \times a^n s_y$ centered on the estimation location is cropped, and its corresponding convolutional feature is extracted. Here, a is the scale factor. Since data with a fixed size is necessary for the dot-product in KCF, we resize each image patch into s_T by means of bilinear interpolation. The final response is calculated by Eq. (4) to obtain the largest value for $\arg\max f(\mathbf{z}^{J_n})$, where J_n has already been resized to the fixed size s_T.

By combining position prediction and scale estimation, the proposed tracker can not only enhance accuracy, but also deal with scale variation.



(a) Occlusion

(b) No update

(c) No occlusion

(d) Update

Figure 2: The first column are the images of sequence tiger from OTB-2015, where the red bounding boxes indicate the tracking object. the response maps in the second column are corresponding to the tracking object. When the object is severely occluded (a), the response map fluctuates intensely and the peak value of the response map is not clear enough (b). When the object is not occluded (c), the peak value of the response map is sharp and clear (d).

2.3 Model Update

To keep the tracker robust, tracking model online updating is very important for the tracking algorithm. Most existed trackers update the tracking models at each frame by using predicted result got from last frame without considering whether the result is accurate or not (Danelljan, 2014), (Ning, 2016). This may cause a deterministic failure once the object is severely occluded or totally missed in the current frame. In our study, we utilize the average of five tracking results with highest peak value to update tracking model, which is named multi-model update strategy.

The peak value of the response map got by KCF can reveal the confidence degree about the tracking results to some extent. The ideal response map should have only one sharp peak. The higher the correlation peak value is, the better the location accuracy is. It can be seen in Figure 2. For model update, a pool with five tracking results from previous frames is fixed, where each result has only sharp peak and highest peak value. When new result is obtained, if it is satisfied with only one sharp peak and its peak value is larger than any of them in the pool, the pool will be updated by deleting the result with smallest peak value and adding the new one.

Then the proposed tracking model will be updated online as follows

Algorithm 1: Proposed tracking algorithm

Variables with hat ^ are in the frequency domain.

- w_sz: size of the tracked region
- *pos*: center location of the tracker in spatial domain
- *patch*: region of *img* centered at *pos* with size w_sz
- *features*(x): extracted from convolutional layers of VGG
- *cos_window*: cosine window weights each feature channel

```
repeat
  if not first image:
    patch            ← region (img, pos, w_sz, motion model)
    features(patch)  ← integrate convolutional features
    $\hat{\mathbf{z}}$                ← F(features(patch) $\odot$ cos_window)
    $\mathbf{k}^{\mathbf{xz}}$              ← correlation( $\hat{\mathbf{z}}$ , $\hat{\mathbf{x}}$ )          ▷ Eq.(5)
    pos              ← $pos + \arg\max(f(\hat{\mathbf{k}}^{\mathbf{xz}}))$     ▷ Eq.(4)
    for every $n$ in $\left\{\left\lfloor -\frac{N-1}{2} \right\rfloor, \ldots, \left\lfloor \frac{N-1}{2} \right\rfloor\right\}$ :
      patch            ← $J_n$ resize to $s_T$
      features(patch)  ← feature extracted
      scale            ← $\arg\max f(\mathbf{z}^{J_n})$
    end for
    $\hat{\boldsymbol{\alpha}}$                ← $\frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}'} + \lambda}$                ▷ Eq.(3)
  Update  tracking model                              ▷ Eq.(7)
until  End of video sequences
```

$$\hat{\alpha} = \sum_{i=1}^{5} \eta_i \hat{\alpha}_i$$
$$\eta_i = \frac{\delta_i}{\sum_{i=1}^{5} \delta_i} \tag{7}$$

where η is the weight for each element in the pool, it is calculated based on peak value $\delta = \arg\max(f(\hat{\mathrm{k}}^{\mathbf{xz}}))$.

An overview of the proposed method is summarized in Algorithm 1.

3 EXPERIMENT RESULTS

In order to evaluate our method's performance, we carry out experiments on the OTB-100 (Wu, 2015) benchmarks. It covers 11 types of challenge scenarios including illumination variation (IV), scale variance (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 clutters (BC) and low resolution (LR). We use one-pass evaluation (OPE) matrix as suggested in (Wu, 2015) to assess our method. Furthermore, to analyze the effectiveness of motion model and model update strategy, we test different version of proposed method on OTB-100. We implemented our method in MATLAB on an Intel i7-4770K 3.50GHz CPU with 24GB RAM, and it can arrive 4 frames per second.

Table 1: Comparison of tracking results of SKCF, SKCF-Motion, SKCF-Multi-model and SKCF-MM.

Tracker	Precision	Success
SKCF	0.840	0.578
SKCF-MOTION	0.851	0.603
SKCF-Multi-model	0.857	0.615
SKCF-MM	0.868	0.639

3.1 Evaluation on Different Version Proposed Method

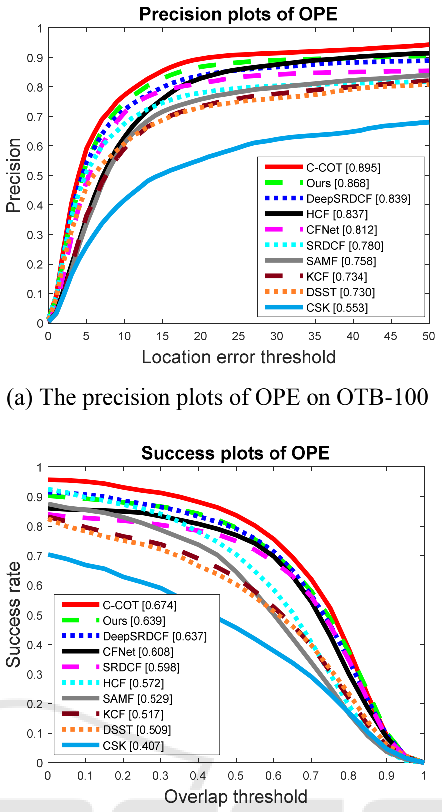
To demonstrate the effect of motion model and multi-model update strategy, we denote the algorithm without motion model and with traditional model update strategy as SKCF, with motion model and traditional model update strategy as SKCF-Motion, without motion model and with multi-model update strategy as SKCF-Multi-model and with both motion model and multi-model update strategy as SKCF-MM. The characteristics and tracking results are summarized in Table 1.

As shown in Table 1, SKCF-MM shows the best tracking accuracy and robustness. Without motion model, SKCF-Multi-model gets poor performance because of fast motion of the object. with traditional model update strategy, the appearance model update in every frame, the tracking bounding box may drift because of occlusion. As shown in experimental results, both motion model and multi-model update strategy improve the tracking performance observably.

3.2 Evaluation on SKCF-MM

We evaluate SKCF-MM with 9 state-of-the-art trackers including KCF (Henriques, 2015), HCF (Ma, 2015), DeepSRDCF (Danelljan, 2015), C-COT (Danelljan, 2016), SAMF (Li, 2014), DSST (Danelljan, 2014), CFNet (Valmadre, 2017), SRDCF (Danelljan, 2015) and CSK (Henriques, 2012).

Figure 3 shows the performance of SKCF-MM with other 9 trackers including top 5 are deep learning-based methods. Although the performance of SKCF-MM is lower than C-COT, the tracking speed is 16 times faster than it which is 0.25 FPS. Our method's effective performance can be explained by three major factors. Firstly, motion model can solve fast motion problem. Secondly, our model update strategy can reduce the influence of occlusion. Thirdly, scale estimation is performed after locating the object, which improves accuracy.



(a) The precision plots of OPE on OTB-100

(b) The success plots of OPE on OTB-100

Figure 3: The precision and success plot of OPE on OTB-100. The numbers on the legend indicate the average precision scores for the precision plot and the average AUC scores for success plot.

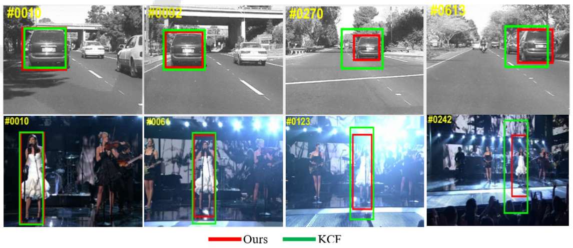


Figure 4: Tracking results over two video sequences with scale variation.

3.3 Scale Variation Evaluation

To demonstrate the effectiveness of our proposed algorithm on scale variation, two typical sequences reflecting scale variation are shown in figure 4, which represents our method and KCF. From the figure, we can observe that our method, which is developed by KCF, can be adaptive to scale variation.

4 CONCLUSIONS

In this paper, we present a novel means of solving the problems of scale variation and appearance model representation. We presented empirical results of different version based on our method, in which we measured the quantitative performance of them. These results demonstrate that motion model and multi-candidate model update strategy can largely improve the algorithm's performance.

Furthermore, the scale variation problem is addressed by means of proposed adaptive scale approach. Moreover, we presented empirical results of various challenging video clips, in which we measured the quantitative performance of our tracker in comparison with a number of state-of-the-art algorithms. Sufficient evaluations on challenging benchmark datasets demonstrate that SKCF-MM tracking algorithm performs well against most state-of-the-art correlation filter-based methods.

ACKNOWLEDGEMENTS

This work was financially supported by the Industrial Strategic Technology Development program of MOTIE/KEIT [10077445, Development of SoC technology based on Spiking Neural Cell for smart mobile and IoT Devices.

REFERENCES

Yilmza, A., Javed, O., and Shah, M., 2006. “Object tracking: a survey,” *ACM Comput. Surv.*

David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming Hsuan Yang, 2008. “Incremental learning for robust visual tracking,” *International Journal of Computer Vision*.

Arnold W. M. Smeulders, Dung M. Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah, 2014. “Visual Tracking: An Experimental Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Babenko, B., Yang, M.-H., and Belongie, S., 2009. “Visual tracking with online multiple instance learning,” *IEEE Conference on Computer Vision and Pattern Recognition*.

Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M. M., Hicks, S. L., and Torr, P. H. Struck, 2016. “Structured output tracking with kernels,” *IEEE transactions on pattern analysis and machine intelligence*.

Zhang J., Ma S., and Stan Sclaroff., 2014. “MEEM: robust tracking via multiple experts using entropy minimization,” *European Conference on Computer Vision*.

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J., 2015. “High-speed tracking with kernelized correlation filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligenc*e.

Ruan, Y., and Wei Z., 2016. “Extended kernelised correlation filter tracking,” *Electronics Letters.*

Li Y., and Zhu J., 2014. “A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration,” *European Conference on Computer Vision Workshops.*

Wang L., Ouyang W., Wang X., and Lu H., 2015. “Visual tracking with fully convolutional networks,” IEEE *International Conference on Computer Vision.*

Simonyan, K., and Zisserman A., 2014. “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556.*

Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O. and Torr, P.H., 2016. “Staple: Complementary learners for real-time tracking,” *IEEE Conference on Computer Vision and Pattern Recognition*.

Danelljan, M., Häger, G., Khan, F., & Felsberg, M., 2014. “Accurate scale estimation for robust visual tracking,” *British Machine Vision Conference.*

Ning J., Yang J., Jiang S., Zhang L., and Yang M.-H., 2016. “Object tracking via dual linear structured SVM and explicit feature map,” *IEEE Conference on Computer Vision and Pattern Recognition*.

Wu, Y., Lim, J., and Yang, M. H., 2015. “Object tracking benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

Ma, C., Huang, J. B., Yang, X. and Yang, M. H., 2015. “Hierarchical convolutional features for visual tracking,” *IEEE International Conference on Computer Vision.*

Danelljan, M., Hager, G., Shahbaz Khan, F., & Felsberg, M., 2015. “Convolutional features for correlation filter based visual tracking,” *IEEE International Conference on Computer Vision Workshops*.

Danelljan, M., Robinson, A., Khan, F. S., & Felsberg, M., 2016. “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” *European Conference on Computer Vision.*

Valmadre, J., Bertinetto, L., Henriques, J. F., Vedaldi, A., and Torr, P. H., 2017. “End-to-end representation learning for Correlation Filter based tracking,” *arXiv preprint arXiv:1704.06036.*

Danelljan, M., Hager, G., Shahbaz Khan, F., and Felsberg, M., 2015. “Learning spatially regularized correlation filters for visual tracking,” *IEEE International Conference on Computer Vision.*

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J., 2012. “Exploiting the circulant structure of tracking-by-detection with kernels,” *European conference on computer vision*.