Scale Invariant Kernelized Correlation Filter based on Gaussian Output

Xiangbo Su, Baochang Zhang, Linlin Yang, and Zhigang Li

School of Automation Science and Electrical Engineering, Beihang University
bczhang@buaa.edu.cn

Abstract. Kernelized Correlation Filter (KCF) is one of state-of-the-art trackers. However, KCF suffers from the drifting problem due to inaccurate localization caused by the scale variation and wrong candidate selection. In this paper, we propose a new method, named Scale Invariant KCF (SIKCF), which estimates an accurate scale and models the distribution of correlation response to address the template drifting problem. The features of SIKCF consist in: 1) A scale estimation method is used to find an accurate candidate. 2) The correlation response of the target image is reasonably considered to follow a Gaussian distribution, which is used to select the better candidate in tracking procedure. Extensive experiments on the commonly used tracking benchmark show that the proposed method significantly improves the performance of KCF, and achieves a better performance than state-of-the-art trackers.

Keywords: Tracking, correlation filters, online learning

1 Introduction

Visual object tracking is one of the most important problem in numerous applications of computer vision, including robotics, video surveillance, and intelligent vehicles [1, 2]. The problem involves estimating the states of the target object in subsequent image frames, with initial state (position and size) given. Despite many works have solved object tracking problem in simple scenes, online object tracking in complex scenarios remains a great challenge due to factors such as illumination change, occlusion, motion blur, texture variation and out-of-view [1–4]. As the appearances of object and background are ordinarily dynamic and variable, the conventional data association and temporal filters [33] relying on motion modeling typically fail.

Most recently, kernelized correlation filters (KCF), which aims to construct discriminative appearance model for tracking from a learning-based perspective, has shown to be promising to handle the appearance variations [11, 6, 12]. The KCF experimentally outperforms many other state-of-the-art trackers while maintaining running at high speed. Despite the performance of the KCF is extremely competitive, it is still prone to drift in long-term tracking, due to the scale variation and wrong candidate selection. The fixed-size search window would introduce noises into the KCF when scale variations exist, which would lead to inaccurate localization. Moreover, the KCF persistently execute its running average procedure without filtrating candidates in updating target appearance, which might introduce even more noises to the filter and eventually leads

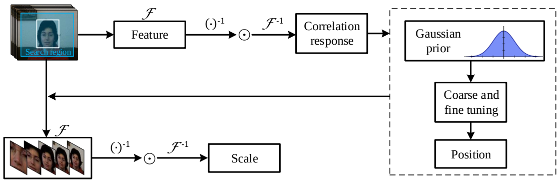

Fig. 1. A scheme of the proposed SIKCF for object tracking.

to drift the tracker away [32, 29]. In this paper, we tackle the problem of template drift to achieve stable long-term tracking and robustness to appearance variation.

Our approach builds on two significant observation. First, we propose an assumption that the maximum correlation response should comply with Gaussian distribution when the target is tracked precisely in each frame. This could be confirmed through observation in practice. Different with existing works which use fixed threshold of correlation output to detect the failure case, we develop a Gaussian constraint method to reduce noisy samples. It is well known that data lies on specific distributions, *i.e.*, faces are considered to be from subspace [30, 31]. As long as the optimal solution resides on the data domain, the constraints derived from the data structure can bring robustness to the variations [16, 15]. Imposing a data structure (distribution) as a constraint is actually a new and flexible way to solve the optimization problems [16, 15], which has been promising in various learning algorithms.

Second, there are two main crux to improve the performance of tracking: (i) estimating the accurate scale of a target and (ii) avoiding from template drift in long-term tracking. For solving the problem of drift, we propose a coarse and fine tuning method based on Gaussian distributed output. The failure cases are detected by determine whether the correlation response comply with a Gaussian distribution. If not, it is reasonable to believe that the tracker drift in current frame. Hence, coarse and fine tuning is executed to detect the translation of the target around. For scale estimation, we learn another separate 1-dimensional correlation filter. We construct a scale feature pyramid as input for training the scale filter, and detect the optimal scale of the target from corresponding response output.

Fig. 1 shows the proposed approach, which mainly innovates at learning robust kernelized correlation filters for object tracking. A key innovation we contribute is to constrain the correlation output to follow a Gaussian distribution, and estimate the accurate scale simultaneously. The Gaussian prior constraint is exploited to model the filter response and reduce noisy samples. By the constraint and scale estimation, our tracker achieve to be stable and robust to scale variance in long-term tracking. Experimentally, we show that our proposed algorithm outperforms state-of-the-art trackers on a benchmark dataset of 51 sequences, while maintaining computational efficient.

2 Related work: Kernelized Correlation Filter

KCF starts from the kernel ridge regression method [11], which is formulated as:

$$\begin{aligned} &\min_{w,\xi} \quad && \sum_i \xi_i^2 \\ &\textit{subject to} && y_i - \boldsymbol{w^T}\boldsymbol{\phi}(x_i) = \xi_i \ \ \forall i; \quad ||w|| \leq B, \end{aligned} \tag{P1}$$

where x_i is the $M \times N$-sized image. $\phi(.)$ is a non-linear transformation. $\phi(x_i)$ (later ϕ_i) and y_i are the input and output of the filter respectively. ξ_i is a slack variable. B is a small constant. According to the Lagrangian method, the objective corresponding to P1 is rewritten as:

$$\mathcal{L}_p = \sum_{i=1}^{M\times N} \xi_i^2 + \sum_{i=1}^{M\times N} \beta_i \left[y_i - \boldsymbol{w}^T\phi_i - \xi_i\right] + \lambda(\|\boldsymbol{w}\|^2 - B^2), \tag{1}$$

where λ is a regularization parameter ($\lambda \geq 0$). From Eq. 1, we have:

$$\begin{aligned} \boldsymbol{\alpha} &= (K + \lambda \boldsymbol{I})^{-1}\boldsymbol{y}, \\ \boldsymbol{w} &= \sum_i \alpha_i \phi_i. \end{aligned} \tag{2}$$

The matrix K with elements $K_{ij} = K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$. Taking advantage of the circulant matrix [10], circulant matrix K can be obtained with circulant inputs and the FFT of $\boldsymbol{\alpha}$ denoted by $\mathcal{F}(\boldsymbol{\alpha})$ is calculated by:

$$\mathcal{F}(\boldsymbol{\alpha}) = \frac{\mathcal{F}(y)}{\mathcal{F}(k^{xx}) + \lambda}, \tag{3}$$

where $\mathcal{F}$ denotes the discrete Fourier operator, and k^{xx} is the first row of the circulant matrix K. In tracking, all candidate patches that are cyclic shifts of test patch z are evaluated by:

$$\mathcal{F}(z) = \mathcal{F}(k^{z\hat{x}}) \odot \mathcal{F}(\boldsymbol{\alpha}), \tag{4}$$

where $\odot$ is the element-wise product and $\hat{x}$ means the target appearance of x. Here $\hat{x}$ is calculated by Eq. 6a [32] and $\mathcal{F}(z)$ is the output response for all the testing patches in frequency domain. Then we have:

$$\hat{\boldsymbol{y}} = \max(\mathcal{F}^{-1}(z)), \tag{5}$$

where $\mathcal{F}^{-1}$ is the inverse FFT operator. The optimal target position is the one with the maximal value among $\hat{\boldsymbol{y}}$ calculated by Eq.5. The target appearance and correlation filter are then updated with a learning rate η as:

$$\begin{cases} \hat{x}^t = (1-\eta)\hat{x}^{t-1} + \eta x^t, & \text{(6a)} \\ \mathcal{F}(\boldsymbol{\alpha^t}) = (1-\eta)\mathcal{F}(\boldsymbol{\alpha^{t-1}}) + \eta\mathcal{F}(\boldsymbol{\alpha}). & \text{(6b)} \end{cases}$$

We use HOG features to learn the translation filter. As an input for the filter, feature map of x_i is extracted from the target region. The position of the target in a new frame is estimated by compute the maximum correlation response $\hat{y}$ using Eq.4 and Eq.5. Then the filter is updated using Eq.6. The following subsection describes how the coarse and fine tuning works based on gaussian constraints.

3 SIKCF

For the purpose of developing an online object tracker that is robust to appearance variation without drift, we divide the task into two parts. The first part is target detection, which locates the target precisely by coarse and fine tuning with gaussian constraints based on KCF. The second part is scale estimation which is achieved by learning a 1-dimensional correlation filter. By this strategy, we can achieve our purpose while still maintaining computational efficient.

3.1 Coarse and Fine Tuning Based on Gaussian Constraints

Since the appearance of a target would alter over time, the tracker might gradually drift and finally fail to track the object. Existing works use fixed threshold of correlation output to detect the failure case and take measure to re-detect the target. That might be not adaptable to all cases since the performance is partly determined by the threshold. We propose an assumption that the maximum correlation response should comply with Gaussian distribution when the target is tracked precisely in every frame. This could be confirmed through observation in practice. we argue that the property of Gaussian prior can well prevent drifting. A sample is adopted only when the maximal response value between the sample and the filter belongs to a Gaussian distribution:

$$\left| \frac{\hat{\boldsymbol{y}}^t - \mu^t}{\sigma^t} \right| < \mathcal{T}_g, \tag{7}$$

where μ^t and σ^t are the mean and variance of the Gaussian distribution respectively in current frame. $\mathcal{T}_g = 1.6$ is empirically set to a constant.

Here we introduce a fine-tuning process to precisely localize the target for sample selection in a local region, instead of searching over the whole image extensively. The tracker activates the fine-tune process when the maximal correlation response is out of the Gaussian distribution (drifting). We first search a coarse region from n_t directions around the latest location (x_0, y_0). The coordinates of a center location for coarse regions are calculated by:

$$p_x = \begin{cases} x_0 + i_r * r_s * cos(i_t * t_s) & \text{for } i_t \text{ mod 2=0} \\ x_0 + i_r * r_s * cos(i_t * t_s + \phi) & \text{for } i_t \text{ mod 2=1,} \end{cases} \tag{8}$$

$$p_y = \begin{cases} y_0 + i_r * r_s * sin(i_t * t_s) & \text{for } i_t \text{ mod 2=0} \\ y_0 + i_r * r_s * sin(i_t * t_s + \phi) & \text{for } i_t \text{ mod 2 =1.} \end{cases} \tag{9}$$

where $r_s = \frac{radius}{n_r}, i_r \in \{1, ..., n_r\}, t_s = \frac{2\pi}{n_t}, i_t \in \{1, ..., n_t\}, \phi = \frac{t_s}{2}$. Finally, $n_r * n_t$ patches centered around the target are cropped as:

$$Z = \{z_1, z_2, , , z_{n_r * n_t}\}. \tag{10}$$

In the coarse process, the maximal correlation response of each patch is obtained by:

$$\begin{cases} r_i = \max(\mathcal{F}^{-1}(\mathcal{F}(z_i))), \\ \mathcal{F}(z_i) = \mathcal{F}(k^{z_i \hat{x}}) \odot \mathcal{F}(\boldsymbol{\alpha}). \end{cases} \tag{11}$$

Then the patch in which the target appears with maximum probability is calculated as:

$$\hat{z} = \underset{i=1}{\overset{n_r * n_t}{\operatorname{argmax}}}(z_i). \tag{12}$$

The fine-tuning step is executed to find the location ($\hat{z}$) of the object precisely as shown in Eq.(4).

Algorithm 1 - The SIKCF algorithm for object tracking

Initial target bounding box $\boldsymbol{b}_0 = [x_0, y_0, w, h]$,
if the frame $n \leq 20$ **then**
    repeat
        Crop out the search windows according to $\boldsymbol{b}_{n-1}$, and extract the HOG features.
        Compute the maximum correlation response $\hat{\boldsymbol{y}}$ using Eq.4 and Eq.5 and record the maximal correlation response as y_n
        The position is obtained according to the maximal correlation response
        Updating target appearance and correlation filter using Eq.6.
    until $n == 20$
end if
Compute the mean μ and variance σ^2 using all previous frames.
if $n > 20$ **then**
    repeat
        Crop out the search window and extract the HOG features.
        Compute the maximal correlation response $\hat{\boldsymbol{y}}$ using Eq.4 and Eq.5.
        if $\left|\frac{\hat{\boldsymbol{y}}-\mu}{\sigma}\right| > \mathcal{T}_g$ **then**
            Crop out the coarse regions $Z = \{z_1, z_2, ..., z_{n_r * n_t}\}$ according to the coordinates calculated by Eq.8 and Eq.9 around the center of $\boldsymbol{b}_{n-1}$
            Coarse searching step: Detect the patch $\hat{z}$ in which the target appears with maximal probability using Eq.11 and Eq.12
            Fine searching step: Locate the object precisely using Eq.4 and Update target appearance and correlation filter using Eq.6.
        end if
        Updating μ and σ^2
        Scale estimation step: Construct the scale pyramid and estimate the optimal scale using Eq.14.Update scale filter using Eq.6.
    until End of the video sequence.
end if

The initialized process is empirically set during the first 20 frames. The fine-tuning strategy is easily implemented to update the localization of the tracked target.

3.2 Scale Estimation

In our approach, the translation of a target is detected by the OCT based kernelized correlation filter, and the scale variation is estimated by learning a separate 1-dimensional correlation filter. Considering that scale variation between two consecutive frames is ordinarily smaller than translation, in a new frame, we first detect the location of target by the OCT kernelized correlation filter. During tracking, we construct a scale feature pyramid as input for training the scale filter, which is consist of features extracted from image patches of various sizes around the estimated position. We consider a target; whose size is $P \times Q$ in current frame. Let N_s be the size of scale filter. For each $k \in \{a^n | n = [-\frac{N_s-1}{2}], [-\frac{N_s-3}{2}], \ldots, [\frac{N_s-1}{2}]\}$, we crop an image patch J_s of size $kP \times kQ$. Here, a donates scale factor. Therefore, N_s image patches with different scale centered around the target are cropped as:

$$J = \{J_1, J_2, ..., J_i, ..., J_{N_s}\}. \tag{13}$$

As in Eq.11, let r_i denote the maximal correlation response of each patch J_i. Then the scale that the target most probably appears with is obtained by:

$$\hat{J} = \operatorname*{argmax}_{i=1}^{N_s}(J_i). \tag{14}$$

Correspondingly, the scale filter is updated separately by Eq.6. To sum up, Algorithm 1 recaps the complete method.

4 Experiments

We evaluate our tracker on a commonly used benchmark [7] that contains 51 sequences, with comparisons to state-of-the-art methods. The selection of Gaussian kernel and most parameters is based on [11]: $\lambda = 10^{-4}$, $\rho = 0.1$. We set the searching window for translation estimation to 1.5 times of the target size. The size of scale space is $N_s = 33$ and the scale factor a is set to 1.02. For other parameters, we empirically set $n_r = 5$, $n_t = 16$ on all sequences. Fig.2a shows that when KCF tracker performs well, the maximal correlation response of the target approximately follows a Gaussian distribution. When KCF fails to track a target, the histogram of response output obviously becomes irregular. In our tracker the output is constrained to comply with a Gaussian distribution, as shown in Fig.2b, meanwhile the tracking performance is significantly improved.

In Fig.3, we report the precision plots which measures the ratio of successful tracking frames whose tracker output is within the given threshold (the x-axis of the plot, in pixels) from the ground-truth, measured by the center distance between bounding boxes. The overall success and precision plots generated by the benchmark toolbox are also reported. These plots report top-10 performing trackers in the benchmarks. As shown in , the proposed method reports the best results. The OCT-KCF and KCF achieve 59.6% and 51.7% based on the average success rate, while the famous Struck and TLD trackers respectively achieve 47.4% and 43.6%. In terms of Precision, OCT-KCF and KCF respectively achieve 80.1% and 74.2% when the threshold is set to 20. OCT-KCF consistently achieves much higher tracking performance than Struck

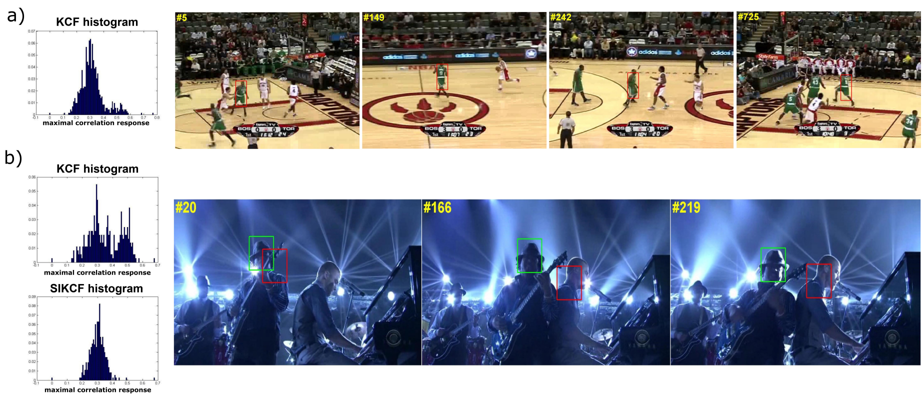

Fig. 2. Illustration of KCF and our algorithm on basketball and shaking sequences. a) A good performance is achieved when the response(output) of KCF is observed to follow a Gaussian distribution on the basketball sequence. b) Our algorithm (green rectangular) is used to improve the performance of KCF (red rectangular) on the shaking sequence, and correlation response in our algorithm follows a Gaussian distribution.

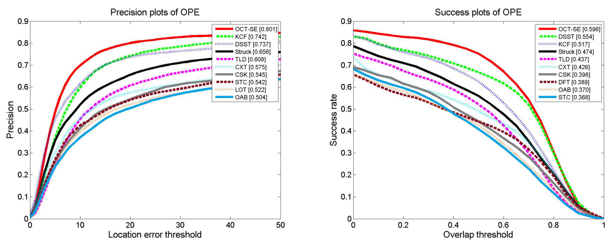

Fig. 3. Success and precision plots according to the online tracking benchmark [7]

(59.8%) and TLD (56.9%). We also compare with DSST, one of latest variants of KCF, which shows that OCT-KCF achieves a significant performance improvement in terms of precision (6.4% improved) and success rate (4.2% improved). These results confirm that the Gaussian prior constraint model contributes to our tracker and enable it to perform better than state-of-the-art trackers. The full set of plots generated by the benchmark toolbox are also reported in Fig. **??** . From the experimental results, it can be seen that the proposed OCT-KCF achieves significantly higher performance in cases of in-plane rotation (5.8% improvement over KCF), scale variations (3.7% improvement over KCF), deformations (6.9% improvement over KCF), out of view (4.6% improvement over KCF) than other trackers (*i.e.*, KCF). This shows that the distribution constrained tracker is more robust to variations mentioned above.

In Fig. 4, we illustrate tracking results from some key frames. In the first row, OCT-KCF can precisely track the coke, while the conventional KCF tracker fails to do that. The famous TLD tracker could relocate the coke target after missing it in 44^{th} frame.

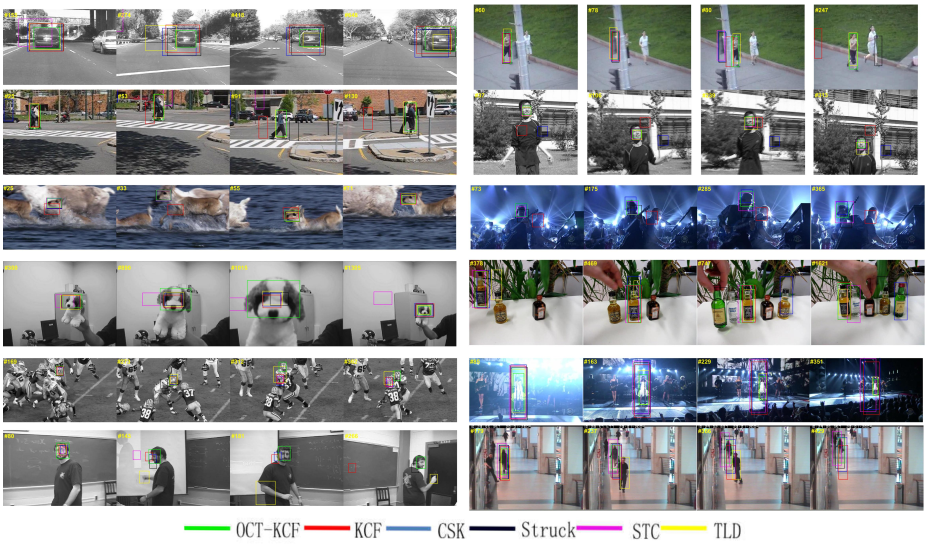

Fig. 4. Illustration of some key frames.

Nevertheless the tracking bounding boxes of the TLD tracker is not as precise as those of OCT-KCF. It is also observed that our proposed OCT-KCF tracker works very well in other sequences, e.g., couple, deer, and football. In contrast, all other compared trackers get false or imprecise results in one sequence at least. On an Intel I5 3.2 GHZ (4 cores) CPU and 8G RAM, the KCF can run up to 185 FPS, while the OCT-KCF achieves 46 FPS. Without losing the real-time performance, the tracking performance is significantly improved by OCT-KCF about 8% on the average success rate and 6% on the precision.

5 Conclusion

In this paper, we propose a stable and robust algorithm for visual object tracking. Our approach learns discriminative correlation filters separately for translation and scale of the target. Estimation for translation is achieved by correlation output constraint to Gaussian distribution based on correlation filter tracking. By coarse and fine tuning, we can precisely estimate the location of target. The optimal scale of target is obtained by learning a 1-dimensional correlation filer, where scale feature pyramid is extracted as the input. Experimentally, we show that the proposed algorithm remarkably improves the performance of KCF, and outperforms the state-of-the-art trackers while still maintaining computational efficient.

References

1. A. Yilmaz, O. Javed, M. Shah, “Object tracking: A survey,” *ACM Computer Survey,* vol.38, no.4, 2006.
2. R. Yao, Q. Shi, C. Shen, Y. Zhang, A. V. Hengel, “Part-based visual tracking with online latent structural learning,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.,* pp.25-27, 2013.
3. A. Adam, E. Rivlin, and I. Shimshoni, “Robust fragments-based tracking using the integral histogram,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.,* pp.798-805, 2006.
4. D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, “Incremental learning for robust visual tracking,” *Int. J. Comput. Vis.,* vol.77, pp: 125-141, 2008.
5. K. Zhang, L. Zhang and M.H. Yang, “Real-time Compressive Tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol.36, no.10, pp.2002-2015,2014.
6. K. Zhang, L. Zhang, M. Yang, and D. Zhang, “Fast Tracking via Spatio-Temporal Context Learning,” *Europ Conf. Comput. Vis.,* 2014
7. Y. Wu, J. Lim, M.-H. Yang, “Online object tracking: A benchmark,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.,* pp.2411-2418, 2013.
8. B. Zhuang, H. Lu, Z. Xiao, D. Wang, “Visual Tracking via Discriminative Sparse Similarity Map,” *IEEE Trans. on Image Process.,* vol.23, no.4, pp.1872-1881, 2014.
9. Z. Han, J. Jiao, B. Zhang, Q. Ye and J. Liu, “Visual object tracking via sample-based adaptive sparse representation,” *Pattern Recognit.,* vol.44, no.9, pp.2170-2183, 2011.
10. J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in *Proc. European Conf. Comput. Vis.,* 2012.
11. J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High speed tracking with kernelized correlation filters,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol.37, no.3, pp.583-596 2015.
12. M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, “Accurate scale estimation for robust visual tracking,” in *Proc. of British Machine Vis. Conf.,* 2014.
13. M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, “Adaptive color attributes for real-time visual tracking,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.,* 2014.
14. D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui., “Visual object tracking using adaptive correlation filters,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.,* 2010.
15. Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *IEEE Trans. Pattern Anal. Mach. Intell.,*, pp.1409-1422, 2012.
16. G. Cabanes and Y. Bennani, “Learning topological constraints in self-organizing map,” in *Proc. of ICONIP*, 2010.
17. Bishop, Christopher M., “Pattern Recognition and Machine Learning,” Springer, ISBN 0-387-31073-8, 2006.
18. Amit Adam, Ehud Rivlin and Ilan Shimshoni, “Robust Fragments-based Tracking using the Integral Histogram,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.,* 2006.
19. D.A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” in *Int. J. Comput. Vis.,* vol.77, pp.125-141, 2008.
20. J. Kwon and K. M. Lee, “Visual tracking decomposition,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.,* pp.1269-1276, 2010.
21. W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank and Z. Zhang, “Incremental tensor subspace learning and its applications to foreground segmentation and tracking,” in *Int. J. Comput. Vis.,* vol.91, no.3, pp.303-327, 2011.
22. S. Avidan, “Support vector tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol.26, no.8, pp.1064-1072, 2004.
23. H. Grabner, M. Grabner, and H. Bischof, “Real-time tracking via on-line boosting,” in *Proc. of British Machine Vis. Conf.,* vol.1, pp.47-56, 2006.

24. H. Grabner, M. Grabner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. European Conf. Comput. Vis.,* pp.234-247, 2008.
25. X. Wang, G. Hua, and T. X. Han, "Discriminative tracking by metric learning," in *Proc. European Conf. Comput. Vis.,* pp.200-214, 2010.
26. S. Hare, A. Saffari, and P. Torr, "Struck: Structured output tracking with kernels," in *Proc. of IEEE Int. Conf. Comput. Vis.,* pp.263-270, 2011.
27. Chun-Te Chu, Jenq-Neng Hwang, Hung-I. Pai, Kung-Ming Lan, "Tracking Human Under Occlusion Based on Adaptive Multiple Kernels With Projected Gradients," in *IEEE Trans. Multimedia,* vol.15, no.7, pp.1602-1615, 2013.
28. B.Babenko, M. Yang and S. Belongie, "Robust object tracking with online multiple instance learning, in *IEEE Trans. Pattern Anal. Mach. Intell.,* vol.6, 2011.
29. B. Zhang, Z. Li, Alessandro Perina, Alessio Del Bue, Vittorio Murio, Jianzhuang Liu. "Adaptive Local Movement Modelling for Object Tracking," *Winter Conf. Comput. Vis.,* 2015.
30. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol.31, no.2, pp.210-227, 2009.
31. B. Zhang, A. Perina, V. Murino, A. Del Bue, "Sparse representation classification with manifold constraints transfer," *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.,* 2015.
32. C. Ma, X.g Yang, C. Zhang, and M.-H. Yang, "Long-term Correlation Tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognition,* 2015.
33. K. Nummiaro, E. Koller-Meier, "An adaptive color-based particle filter," *Image and Vision Computing,* vol.21, no.10, pp.99-110, 2003.
34. Hyeonseob Nam , Bohyung Han, Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. Axiv.2015.
35. Jialue Fan, Wei Xu, Ying Wu, Human Tracking Using Convolutional Neural Networks. IEEE Trans. Neural Network,2010.