

Object Tracking Based on Mean Shift Algorithm and Kernelized Correlation Filter Algorithm

Huazheng Zhou^{1,2}, Xiaohu Ma^{1,2(✉)}, and Lina Bian^{1,2}

¹ School of Computer Science and Technology, Soochow University,
Suzhou 215006, China
xhma@suda.edu.cn

² Collaborative Innovation Center of Novel Software Technology
and Industrialization, Nanjing 210023, China

Abstract. In order to solve the problems of motion blur and fast motion, a new robust object tracking algorithm using the Kernelized Correlation Filters (KCF) and the Mean Shift (MS) algorithm, called KCFMS is presented in this paper. The object tracking process can be described as: First, we give the initial position and size of the object and use the Mean Shift algorithm to obtain the position of the object. Second, the Kernelized Correlation Filtering algorithm is used to obtain the position of the object in the same frame. Third, we use the cross update strategy to update the object models. In order to improve the tracking speed as much as possible, our object tracking algorithm works only over one layer. This hybrid algorithm has a good tracking effect on the target fast motion and motion blur. We present extensive experimental results on a number of challenging sequences in terms of efficiency, accuracy and robustness.

Keywords: Kernelized Correlation Filters · Mean shift · Motion blur · Fast motion

1 Introduction

Visual tracking is a fundamental problem in computer vision, which finds a wide range of application areas [1]. Recently, hybrid discriminative generative methods have opened a promising direction to benefit from both types of methods. Several hybrid methods [2–6] have been proposed in many application domains. These methods train a model by optimizing a convex combination of the generative and discriminative log likelihood functions. The improper hybrid of discriminative generative model generates even worse performance than pure generative or discriminative methods. In this paper, our tracking algorithm only gives the initial position and size of the target in the first frame of the validation set GT (represented by a rectangular area). We use a tracking algorithm based on the discriminant and generated model to learn the appearance model. Discriminant methods focus on finding a decision boundary to distinguish between background and goals. Discriminant methods not only focus on the target but also focus on its background. The tracker based on the generation method focuses only

on the target itself. For example, based on the histogram method, these methods are simple, but the tracking effect is very good.

Our hybrid algorithm uses a two-step tracking method. First, we use the mean shift algorithm to predict the position of the target in the current frame, and then use this position to sample. Second, we use the kernel correlation filtering algorithm to determine the position of the target in the frame. As shown in Fig. 1, we use cross update strategy to update the appearance model in the current frame. We use the position obtained by the MS method to update the KCF appearance model. Meanwhile, we use the position obtained by the KCF method to update the MS appearance model. This step occurs at the stage of the model updating.

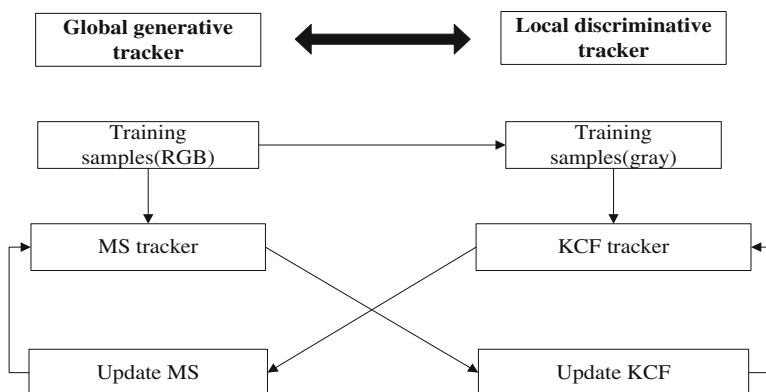


Fig. 1. The appearance models update strategy

In Sect. 2, the related work is briefly reviewed. Our method is proposed in Sect. 3. Section 4 shows the experimental results and analysis. Finally Sect. 5 draws the conclusion.

2 Related Work

A typical failure of visual tracking is drift, which means that tracking performance gradually degrades over time, and eventually the tracker lose the object. There exist several reasons to explain the drift issue. One explanation is that self-learning reinforces previous errors and causes the drift. Tang et al. [7] proposed to use co-training to online train two SVM trackers with color histogram features and HOG features. This method uses an SVM solver [8] to focus on recent appearance variations without representing the global object appearance. Babenko et al. [9] argued that the drift can be also caused by the ambiguities when using online labeled data to update the model. For example, the way of selecting positive and negative samples inevitably contain ambiguities, and these ambiguities lead to an offset of the online trained model. A multiple instance learning approach was introduced to avoid ambiguities by using a bag of samples instead of an individual sample. More recently, Kalal et al. [10],

in another way, proposed a very efficient method using a combination of detection, tracking, and modeling modules. This tracker is robust against drifting by bootstrapping itself using positive and negative constraints.

In this paper, we propose to use the two stage methods to combine generative and discriminative models. That is to say, first we use the mean shift algorithm to predict the position of the target in the current frame, and then use this position to sample, using the kernel correlation filtering algorithm to determine the position of the target in the current frame.

2.1 The Mean Shift Algorithm

The mean shift [11] tracking process consists of two components. These are the target representation and the mean shift iteration. The color histogram is used to represent the target. The target region has n pixels, and the i -th pixel is denoted by $\{x_{ms}^i\}_{i=1,\dots,n}$. The probability of a feature u is an m -color-bin histogram. The target model $q_u (u = 1, 2, \dots, m)$ is computed as follows:

$$q_u = C \sum_{i=1}^n k \left(\left\| \frac{x_{ms}^i - x_{ms}^*}{h} \right\|^2 \right) \delta[b(x_{ms}^i) - u] \quad (1)$$

where x_{ms}^* is the target center, $k(x)$ is the isotropic kernel profile. $\delta(x)$ is the Kronecker delta function, $b(x)$ maps the pixel of a coordinate to feature space and C is the constant:

$$C = 1 / \sum_{i=1}^n k \left(\left\| \frac{x_{ms}^i - x_{ms}^*}{h} \right\|^2 \right) \quad (2)$$

Similarly, the target candidate model from the target candidate region centered at position y is given by $\{p_u(y)\}_{u=1,2,\dots,m}$.

A key issue in the Mean Shift algorithm is the computation of an offset from the current location to a new location. We use the Bhattacharyya coefficients to compute the similarity between the target histogram and the candidate histogram.

$$\rho[p(y), q] = \sum_{u=1}^m \sqrt{p_u(y)q_u} \quad (3)$$

2.2 The Kernelized Correlation Filters Algorithm

The KCF algorithm [12] uses the cyclic shifts to obtain large samples and then uses these samples to train the Classifier. The training process of the classifier can be described by the following formula:

$$\min_w \sum_i (f(X_i) - y_i)^2 + \lambda \|w\|^2 \quad (4)$$

where we need find the best w . We can refer to the paper [12].

$$w = \sum \alpha_i \varphi(X_i) \quad (5)$$

3 Overview of the Proposed Approach

KCF algorithm is difficult to deal with motion blur and fast motion. In order to resolve the drawback of KFC, a target tracking algorithm is proposed to joint kernel correlation filtering and mean shift. In each frame of the video, the hybrid algorithm first uses the mean shift algorithm to predict the target position in the current frame, and then uses this position as the input of the kernel correlation filtering algorithm to detect the target position, and finally uses the cross update strategy to update the target model. In addition, in order to maximize the speed of target tracking, the hybrid tracking algorithm has only one layer.

Our discriminant model uses the kernel correlation filter (KCF) algorithm to generate the model using the mean shift (MS) algorithm. Since our hybrid method is based on these two algorithms, we called our own algorithm as KCFMS. We first read RGB image at the t -th frame, then use MS algorithm to determine the target position l_{ms} . Next, we convert the image to gray image, and use l_{ms} and gray image as the input of kernel correlation filtering algorithm at the t frame. The output of the KCF is the position l_{kcf} . Then we use l_{kcf} as the t frame location l_t . This algorithm is parallel tracking algorithm. The specific process of the hybrid algorithm KCFMS proposed in this chapter is described as follows:

The KCFMS Algorithm.

Input: the number of the frames and the current image

Output: The target position of the t frame

- (1) Read the first frame image and the initial position and size of the target
 - (2) Obtain the generation model by Eq.(1)
 - (3) Training the classifier by Eq.(4)
 - (4) Read a new frame image
 - (5) Determines the target position l_{ms} with the generating method MS,
 - (6) Use the location l_{ms} as the input of the KCF method, then Calculate the position of the prediction target is l_{kcf}
 - (7) Make $l_t = l_{kcf}$, as the estimate of the target position at the t frame.
 - (8) Updating the KCF classifier.
 - (9) If it is not the last frame, go to step (4), otherwise go to step (10).
 - (10) End the target tracking procedure.
-

We also give a flow chart of the algorithm, as shown in Fig. 2. In this flow chart, we have detailed the implementation of the algorithm in this chapter. The left dashed box indicates the work done by the algorithm on the first frame image, and the right side is the tracking flow of the algorithm on the remaining image sequence.

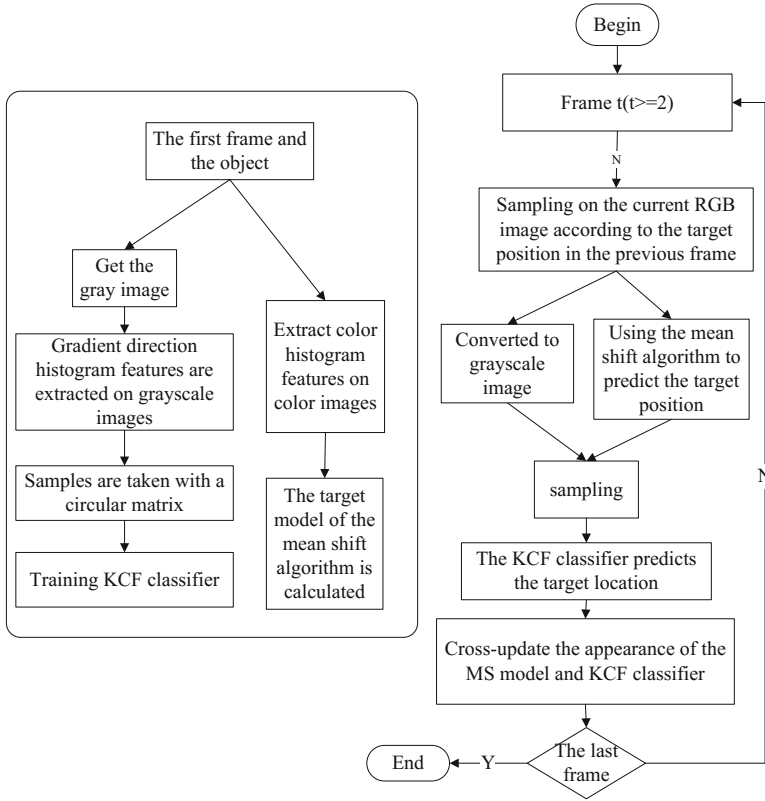


Fig. 2. The flowchart of our proposed algorithm

4 Experiments and Discussion

The experiment inputs the position and the size of the target in the first frame, and then tracks the target in the remaining image sequence. In order to fully analyze the performance of the algorithm in this paper, our experiment is carried out on 50 data sets [13], and the algorithm is analyzed comprehensively from qualitative analysis and quantitative analysis. In addition, the running environment is Windows10 operating system, and programming platform is Matlab2013a.

Our algorithm compares with the other four main algorithms. These algorithms are FCT [14], MIL [9], STC [15], and KCF [12], respectively. It is very likely that the target tracking algorithm based on the hybrid model have no better tracking effect than

the single algorithm, while the kernel correlation filtering algorithm is closely related to the algorithm of our paper. Therefore, our algorithm must be compared with the KCF algorithm. For the mean shift algorithm MS is quite poor, so the article is not necessary to be compared with this algorithm. In addition, our algorithm has only one layer. In this paper, we use the average center location error and the overall the average center location error to measure the performance of the algorithm.

4.1 Qualitative Comparison

Qualitative comparison gives the most intuitive description of the tracking results, and we can see the tracking result of each algorithm on the same data set. Qualitative comparison shows two data sets (David3 and Basketball) in the 50 datasets [13], which are used to test the ability of the algorithm, and thus improving the understanding of our algorithm. We will analyze the characteristics of the five data sets of Couple, David3, Boy, DragonBaby, and Girl2, as shown in Table 1. Then we analyze the intuitive tracking effect of each algorithm in the two data sets.

Table 1. Partial data set and characteristics

Sequence	Frames	Data characteristics
Couple	140	Scale Variation, Deformation, Fast Motion, Background Clutters
David3	252	Occlusion, Deformation, Rotation, Background Clutters
Boy	602	Scale Variation, Motion Blur, Fast Motion, Rotation
DragonBaby	113	Scale Variation, Occlusion, Motion Blur, Fast Motion, Rotation, Out-of-View
Girl2	1500	Scale Variation, Occlusion, Deformation, Motion Blur, Deformation

On the David3 dataset, the five algorithms tracked David. The dataset shows that David's journey is back and forth in the outdoor scene, with occlusion and rotation on the way. In general, our algorithm tracks the target throughout the process. Especially in the 83rd and the 187th frame when the object is occluded heavily by the tree, the KCFMS algorithm also can track the target. In the 139th frame around the target begin to go back, that is to say, the target appearance change, but our algorithm can track the object successfully. Part of the details shown in Fig. 3.

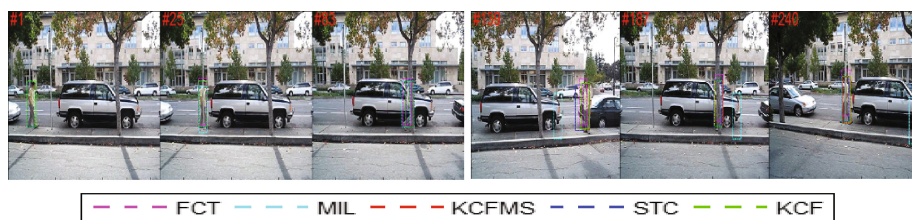


Fig. 3. Partial results of the five algorithms on the David3 dataset

Basketball is a complex data set with fast motion, target deformation, occlusion from similar object, illumination variation, and so on. Therefore, it is a data set favored by researchers of many target tracking algorithms. The result indicates that most of the algorithm’s tracking results were good before the 230th frame, and only the MIL algorithm drifted slightly in the 290th frame due to the approximation of similar object. In the same time, the STC algorithm is drift. This data set also has a noticeable change in illumination variation, and we give the light changes in the 650th frame. In order to see the changes in illumination variation, we give the frame before and after the two tracking effect. We can see that our algorithm’s tracking effect is very good in this case. In addition, the data set itself has low pixel. The details are shown in Fig. 4.

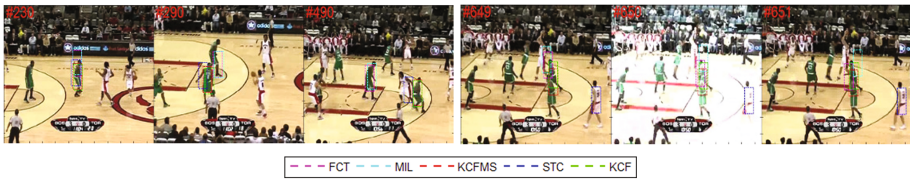


Fig. 4. Partial results of the five algorithms on the Basketball dataset

4.2 Quantitative Comparison

We analyze our algorithm for quantitative comparison from two aspects, the first is the average center error, and the other is the sum of error and mean error. Table 2 shows the overall mean center error values for each algorithm over 50 datasets, which is a measure of the target tracking algorithm. In this paper, the best value of the mean center error is marked in boldface, and the second mean center error is marked by the italicized slash. It can be seen from the Table 2 in which the tracking result of our algorithm is optimal.

Table 2. All mean center error values for each algorithm on 50 datasets (unit: pixels)

Algorithm	FCT [14]	MIL [9]	KCFMS	STC [15]	KCF [12]
Mean value	45.16	52.62	26.53	71.35	<u>35.98</u>

Table 2 shows the Mean Value of the algorithm, but it is not possible to gain a better understanding of the tracking effect of the algorithm in different datasets. Therefore, we will analyze the tracking result of each algorithm from the three aspects of illumination change (IV), motion blur (MB), fast moving (FM). For the analysis of these three characteristics, this paper uses the center error of each data set, the sum of center error and the average center error of these three aspects to analyze the tracking effect of the algorithm.

Table 3 gives the center errors for the eight data sets with the characteristics of the illumination change IV. In addition, we give the sum of the errors and calculate the mean value of the error. From Table 3, although the tracking results of the algorithm in each data set are not always good, the optimal tracking result is less than the KCF algorithm and the overall central error is the smallest which is 22.36 Pixels. Therefore, our algorithm is robustness to the illumination change.

Table 3. Average center location error with IV feature data set (unit: pixels)

Algorithm	FCT [14]	MIL [9]	KCFMS	STC [15]	KCF [12]
Basketball	88.52	103.80	7.18	78.66	<u>7.89</u>
David	29.35	44.83	6.28	36.35	<u>8.06</u>
Doll	27.15	22.41	<u>12.19</u>	140.15	8.36
FaceOcc2	22.39	17.62	<u>7.87</u>	18.57	7.67
Fish	8.91	12.68	7.61	<u>4.50</u>	4.08
Human7	18.19	32.98	30.89	<u>24.93</u>	48.20
Shaking	<u>27.61</u>	164.38	93.65	16.14	112.50
Tiger1	24.89	103.15	<u>13.21</u>	63.08	8.05
the Sum of Error	247.00	501.85	178.88	382.39	<u>204.80</u>
Mean Error	30.86	62.73	22.36	47.80	<u>25.60</u>

Table 4 shows the tracking results for each algorithm on a motion set with motion fuzzy MB. First, we can see from the algorithm that the average center error on each data set is optimal. Because in this 11 data set on our algorithm has four best (bold font representation) and three times (italic underlined) of the tracking results. Second, from the last two lines of data in Table 4, we can see that our algorithm error is 37.23 pixels, which is far better than the other four algorithms. We can also see that the STC algorithm has the second tracking effect, so we can deduce that the algorithm is very good for the target tracking effect in the case of occlusion.

Table 4. Average center location error with MB feature data set (unit: pixels)

Algorithm	FCT [14]	MIL [9]	KCFMS	STC [15]	KCF [12]
Biker	<u>27.99</u>	16.63	44.23	82.87	77.18
BlurCar3	180.53	90.46	<u>18.44</u>	52.09	4.14
BlurOwl	<u>173.18</u>	190.20	14.34	240.64	183.43
Box	104.03	28.94	118.35	100.56	<u>89.13</u>
Boy	8.99	15.97	2.70	18.30	<u>2.87</u>
DragonBaby	<u>37.89</u>	74.70	16.08	174.20	50.40
Girl2	<u>108.37</u>	172.37	51.12	267.49	264.58
Human7	18.19	32.98	30.89	<u>24.93</u>	48.20
Jumping	46.71	12.85	<u>18.78</u>	93.70	26.12
Tiger1	24.89	103.15	<u>13.21</u>	63.08	8.05
Tiger2	34.94	<u>46.83</u>	81.32	201.75	47.44
the Sum of Error	<u>765.71</u>	785.08	409.48	1319.61	801.52
Mean Error	<u>69.61</u>	71.37	37.23	119.96	72.87

Fast motion is also a hot topic in object tracking. Table 6 gives the tracking results of the algorithm on 15 datasets with fast motion. First, we can see that the optimal tracking result of KCF algorithm is the best from the average center error of the algorithm in each data set. The tracking result of STC algorithm is the worst, and the optimal tracking result of the remaining three algorithms is basically same. Then, looking at the overall tracking results, that is the sum of the errors in Table 5 and the mean of the errors. From this we can see that our algorithm is superior to KCF algorithm, KCF algorithm is ranked second, FCT and MIL algorithm are similar, STC algorithm tracking results are still the worst. That is to say, the KCFMS algorithm has better tracking performance than other algorithms for fast motion.

Table 5. Average center location error with FM feature data set (unit: pixels)

Algorithm	FCT [14]	MIL [9]	KCFMS	STC [15]	KCF [12]
Biker	<u>27.99</u>	16.63	44.23	82.87	77.18
Bird2	80.94	15.71	<u>16.02</u>	18.44	21.37
BlurCar3	180.53	90.46	<u>18.44</u>	52.09	4.14
BlurOwl	<u>173.18</u>	190.20	14.34	240.64	183.43
CarScale	26.63	40.04	<u>25.51</u>	96.18	16.14
ClifBar	24.98	<u>34.89</u>	35.10	36.96	36.72
Coke	43.81	75.84	97.08	17.14	<u>18.65</u>
Couple	<u>34.93</u>	38.39	15.32	826.12	47.56
DragonBaby	<u>37.89</u>	74.70	16.08	174.20	50.40
Human7	18.19	32.98	30.89	<u>24.93</u>	48.20
Jumping	46.71	12.85	<u>18.78</u>	93.70	26.12
Surfer	49.20	23.43	<u>10.56</u>	49.74	8.74
Tiger1	24.89	103.15	<u>13.21</u>	63.08	8.05
Tiger2	34.94	<u>46.83</u>	81.32	201.75	47.44
Vase	19.84	22.40	25.18	<u>15.01</u>	12.43
the Sum of Error	824.63	818.51	462.07	1992.85	<u>606.56</u>
Mean Error	54.98	54.58	30.8	132.86	<u>40.44</u>

In summary, the results of Tables 3, 4 and 5 show that the KCFMS algorithm is better than other algorithms in the three aspects of illumination, motion blur, fast motion. It can be seen that the tracking results of the hybrid tracking algorithm is perfect in this paper.

5 Conclusion

This paper presents a hybrid tracking algorithm. We use the MS tracker to track the RGB image, and then the RGB image is converted into the gray image. In the gray image we use the KCF method to track the object. When we obtain the object position, we use the cross update method to update the object appearance model. We do a lot of experiments and give the relevant experimental results in the public data sets.

Comparing with the mainstream of the algorithms is to analyze the tracking result. Experiments show that the proposed algorithm is more stable and has good tracking result on the changes of light, fast motion, motion blur, occlusion, background clutter and scale change. In this paper, the algorithm does not achieve the scale adaptability, but the scale adaptive is also a very important research direction of the object tracking, so our next goal is to achieve the scale of the algorithm.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (61402310). Natural Science Foundation of Jiangsu Province of China (BK20141195).

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. *ACM Comput. Surv.* **38**(4), 1–17 (2006)
2. Lasserre, J.A., Bishop, C.M., Minka, T.P.: Principled hybrids of generative and discriminative models. In: 19th IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 87–94. IEEE Computer Society, New York (2006)
3. Ng, A., Jordan, M.I.: On discriminative vs. generative classifiers: a comparison of logistic regression and Naive Bayes. In: *Proceedings of Advances in Neural Information Processing*, vol. 28, no. 3, pp. 169–187 (2001)
4. Lin, R.S., Ross, D.A., Lim, J., et al.: Adaptive discriminative generative model and its applications. In: *Neural Information Processing Systems*, pp. 801–808 (2004)
5. Yang, M., Wu, Y.: Tracking non-stationary appearances and dynamic feature selection. In: 18th IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1059–1066. IEEE Computer Society, San Diego (2005)
6. Yu, Q., Dinh, T.B., Medioni, G.: Online tracking and reacquisition using co-trained generative and discriminative trackers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008*. LNCS, vol. 5303, pp. 678–691. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-88688-4_50](https://doi.org/10.1007/978-3-540-88688-4_50)
7. Tang, F., Brennan, S., Zhao, Q., et al.: Co-tracking using semi-supervised support vector machines. In: 9th IEEE International Conference on Computer Vision, pp. 1–8. IEEE (2003)
8. Cauwenberghs, G., Poggio, T.: Incremental and decremental support vector machine learning. In: 13th International Conference on Neural Information Processing Systems, vol. 1, pp. 388–394. MIT Press, Denver (2000)
9. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(8), 1619–1632 (2011)
10. Kalal, Z., Matas, J., Mikolajczyk, K.: P-N learning: bootstrapping binary classifiers by structural constraints. In: 23rd IEEE Conference on Computer Vision and Pattern Recognition, vol. 238, pp. 49–56. IEEE Computer Society, San Francisco (2010)
11. Comaniciu, D., Menber, V.R., Meer, P.: Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 564–575 (2003)
12. Henriques, J.F., Rui, C., Martins, P., et al.: High-speed tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(3), 583–596 (2014)
13. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. *IEEE Trans. Comput. Vis. Pattern Recogn.* **37**(9), 1834–1848 (2015)

14. Zhang, K., Zhang, L., Yang, M.H.: Fast compressive tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(10), 2002–2015 (2014)
15. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.-H.: Fast visual tracking via dense spatio-temporal context learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014. LNCS*, vol. 8693, pp. 127–141. Springer, Cham (2014). doi:[10.1007/978-3-319-10602-1_9](https://doi.org/10.1007/978-3-319-10602-1_9)