LETTER

Prediction-Based Scale Adaptive Correlation Filter Tracker

Zuopeng ZHAO[†,††], *Nonmember*, **Hongda ZHANG**[†,††], *Student Member*, **Yi LIU**[†,††a)], **Nana ZHOU**[†,††], **Han ZHENG**[†,††], **Shanyi SUN**[†,††], **Xiaoman LI**[†,††], *and* **Sili XIA**[†,††], *Nonmembers*

SUMMARY Although correlation filter-based trackers have demonstrated excellent performance for visual object tracking, there remain several challenges to be addressed. In this work, we propose a novel tracker based on the correlation filter framework. Traditional trackers face difficulty in accurately adapting to changes in the scale of the target when the target moves quickly. To address this, we suggest a scale adaptive scheme based on prediction scales. We also incorporate a speed-based adaptive model update method to further improve overall tracking performance. Experiments with samples from the OTB100 and KITTI datasets demonstrate that our method outperforms existing state-of-the-art tracking algorithms in fast motion scenes.

key words: *visual tracking, correlation filter, scale prediction, model update, fast motion*

1. Introduction

Visual object tracking, which has applications in a wide range of areas, is a fundamental problem in computer vision. Despite significant progress in recent years, object tracking remains a challenging problem due to uncertain factors, such as scale variation, fast motion, and motion blur.

After the development of Bolme et al.'s MOSSE [1] algorithm, which uses correlation filtering methods for tracking, discriminative correlation filter (DCF)-based methods have gained popularity and have been constantly evolving. In recent years, DCF-based approaches have shown outstanding results in object tracking benchmarks [2], [3]. The improvement in DCF-based tracking performance is mainly due to improvements in feature selection [4], [5], scale estimation [4], [6], [7], and tracking models [5], [8]–[11]. Among them, DRT [10] takes both discrimination and reliability information to reduce the tracking-model degradation caused by the unexpected salient regions on the feature map. Furthermore, ASRCF [11] adopts an adaptive spatial regularization scheme to learn an effective spatial weight for a specific object and its appearance variations, and therefore result in more reliable filter coefficients during the tracking process. Early correlation filter class tracking algorithms, such as CSK [12], and KCF [8], have a fixed template size. When the tracking target scale changes, the tracker is unable to accurately locate the target, causing an excessive amount of interference information to be learned. To solve this problem, a scale pool method is proposed. The SAMF [4] algorithm scales the initial target at seven different levels, calculates the corresponding response values, and selects the target scale value with the largest response. The DSST [6] approach treats target tracking as two independent problems, namely target center translation and target scale change, then proposes a fast scale estimation approach by learning distinct filters for translation and scale. Both algorithms alleviate the impact of scale changes on tracking performance to some extent. However, the algorithms have issues keeping up with the scale changes of fast moving targets. Although increasing the number of layers in the scale pyramid can improve performance, this also increases the time cost and negatively impacts the speed of the correlation filter tracking methods. Furthermore, these methods are limited to tracking using a fixed number of scales. When a scale outside the preset scale range, the filter learns a significant amount of background information, excessive attention, and target local texture.

To overcome these limitations, we propose a scale estimation and adaptive model update approach based on the correlation filter framework. The key contributions of this work can be summarized as follows. First, we add scale prediction to the scale adaptive scheme to extend the capability of the tracker, thereby allowing it to handle scale changes in fast motion situations. Secondly, we use a speed-based model update method to reduce unnecessary learning and improve model robustness. Experiments were performed on fast motion sequences from the OTB100 [3] and KITTI [13] datasets. The results show that our method can achieve better performance on target tracking tasks than existing state-of-the-art trackers in fast motion scenes.

2. The Proposed Approach

Recently, the ECO [14] tracker has achieved excellent performance on different benchmark datasets. Thus, we based our implementation on the ECO tracker.

2.1 The ECO Tracker

The ECO algorithm extracts a D-dimensional feature $x_j^1, \ldots, x_j^D$ from an image patch and introduces an interpo-

Manuscript received May 20, 2019.
Manuscript revised July 8, 2019.
Manuscript publicized July 30, 2019.

[†]The authors are with the School of Computer Science and Technology, CUMT, China.
[††]The authors are with the Mine Digitization Engineering Research Center of Minstry of Education of the People's Republic of China, China.
a) E-mail: ts18170094p21@cumt.edu.cn

DOI: 10.1587/transinf.2019EDL8101

lation function b_d to interpolate the discrete image feature function $x_j^d[n]$ into the continuous domain $[0, T) \subset \mathbb{R}$ to obtain $J_d x^d(t)$.

$$J_d\{x^d\}(t) = \sum_{n=0}^{N_d-1} x^d[n] b_d \left(t - \frac{T}{N_d} n\right) \tag{1}$$

The interpolated samples $J_d x^d(t)$ are superimposed on shifted versions of the interpolation function $b_d \in L(T)$. The feature value $x^d[n]$ is used as the weight of each of these shifted versions. Function b_d is based on a cubic spline kernel $b(t)$.

To reduce the number of model parameters, a factorization convolution method is introduced. Some of the filters learned by C-COT contain negligible energy which contribute very little to locating the target, but their presence affects the training time. Thus, our approach does not learn separate filters for each channel. We use a set of filters $f^1, \ldots, f^c$, where $C < D$. The filter for feature layer d is then constructed as a linear combination $\sum_{c=1}^{C} p_{d,c} f^c$ of the filters f^c using a set of learned coefficients $p_{d,c}$. These coefficients can be expressed compactly as a $D \times C$ matrix $P = (p_{d,c})$. The new multichannel filter can be expressed as Pf. Then, the detection scores of the target can be written as follows:

$$S_{Pf}\{x\} = Pf * J(x) = \sum_{c,d} p_{d,c} f^c * J_d\{x^d\} = f * P^T J\{x\} \tag{2}$$

The last equation is based on the linearity of convolution. Therefore, the factorized convolution can also be viewed in the following manner, in which each position t of the feature vector $J\{x\}(t)$ is first multiplied by the matrix P^T, after which the obtained C-dimensional feature map is convolved with the filter f. In other words, the matrix P^T can be regarded as a linear dimensionality reduction operator.

To avoid the large memory requirements and computational burden caused by the larger set of training samples, the ECO tracker utilizes a probabilistic generative model of the sample set that achieves a compact description of the samples by eliminating redundancy and enhancing variety. Traditional filters are learned by minimizing the following objective:

$$E(f) = \sum_{j=1}^{M} \alpha_j \left\| S_f\{x_j\} - y_j \right\|^2 + \sum_{d=1}^{D} \left\| \omega f^d \right\|^2 \tag{3}$$

Assuming that the joint probability distribution of the sample feature map is x and the expected response score y is $p(x, y)$, the intuitive objective is to find the filter that minimizes the expected correlation error. This is obtained by replacing (3) with

$$E(f) = \mathrm{E}\left\{ \left\| S_f\{x\} - y \right\|^2 \right\} + \sum_{d=1}^{D} \left\| \omega f^d \right\|^2 \tag{4}$$

where $\mathbb{E}$ is the expectation of a random variable subject to the joint probability distribution $p(x, y)$. The original loss function is obtained when $p(x, y) = \sum_{j=1}^{M} \alpha_j \delta_{x_j, y_j}(x, y)$, where δ_{x_j, y_j} represents the Dirac impulse for the training sample (x_j, y_j). Because the expected output scores y_j corresponding to different frames are the same, this can be written as $y_j = y_0$. Thus, the sample distribution can be expressed as $p(x, y) = p(x)\delta_{y0}(y)$. At this stage, it is only necessary to find a more suitable $p(x, y)$. Introducing a Gaussian Mixture Model (GMM), the following is obtained:

$$p(x) = \sum_{l=1}^{L} \pi_l \mathcal{N}(x; \mu_l; I) \tag{5}$$

where L represents the number of Gaussian components $\mathcal{N}(x; \mu_l; I)$, and π_l and μ_l correspond to the weight and mean, respectively, of the component l. Finally, the loss function changes to,

$$E(f) = \sum_{l=1}^{L} \pi_l \left\| S_f\{\mu_l\} - y_0 \right\|^2 + \sum_{d=1}^{D} \left\| \omega f^d \right\|^2 \tag{6}$$

The number of samples is reduced from M to L using the method above. The ECO [14] tracker employs the Gauss-Newton and use the Conjugate Gradient methods to optimize the quadratic subproblems.

2.2 Scale Prediction

As discussed earlier, we adopt the scale adaptive method based on prediction to avoid the problem of scale outside the preset scale range when moving rapidly. Even if the target scale changes drastically, accurate scale estimation can be done using a few scale search regions. Depending on the continuity of the motion of the object, the scale change during the movement may be gradual instead of abrupt. Therefore, the forward difference is used to calculate the scale change and estimate the target scale in the next frame. This can be expressed as follows:

$$S'_{j+1} = S_j + \sum_{h=1}^{H} 2^{-h} \eta (S_{j-h+1} - S_{j-h}) \tag{7}$$

where S_j represents the target scale of frame j, S'_{j+1} is the predicted target scale, H is the memory factor indicating the number of frames affecting the scale prediction, and η is the weight with respect to the rate of change. Considering that the rate of change of the scale of the target motion differs between scenes, if a fixed value of η is used, the tracker may lag when the target's speed suddenly increases or decreases.

To enhance the robustness of the system and improve its response time. The value of η is dynamically adjusted per the principle of classical feedback control, which is used in traditional automation systems. This is shown in Fig. 1. Ordinary control systems can achieve a stable output when there is no interference. In such a scenario, closed-loop control is not required. However, in this application, this type of scenario is impossible. The feedback control system is fully functional when there are unpredictable distur-

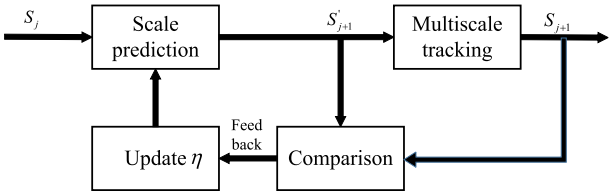

Fig. 1 Flow chart of the scale adaptive scheme based on prediction scales, joint scale rate of change, and feedback control to achieve scale prediction. Scale prediction is added before the common multiscale tracking module, and the value of η is dynamically adjusted according to the difference between the optimal scale S_{j+1} and the prediction scale S'_{j+1} to achieve accurate scale estimation.

bances or unpredictable changes. In each frame, we use $Scale\_change = S_{j+1}/S'_{j+1}$ to characterize the difference between the optimal scale and the prediction scale, the value of $\eta = \eta Scale\_change$, and the closed-loop control scheme is continuously corrected until it attains a steady state. Therefore, it follows that the tracker should be able to adapt to fast motion scenes.

2.3 Adaptive Model Update

To increase the accuracy of the location, the DCF tracker updates the filter model every frame. However, in recent work, it was demonstrated that ECO and Siamese networks were found to perform well with infrequent updates to the model. Therefore, we argue that excessive updates is unnecessary.

The optimization process should only begin when the appearance of the target exhibits sufficient change. However, determining how much change is sufficient is not a straightforward process, and may require complex and time-consuming calculations. Inspired by the ALRCF [15] algorithm, we argue that the necessary change in appearance depends on the velocity of the target, defined as the pixel difference between the present and previous positions of the object in each frame. Therefore, instead of ECO [14] using a fixed update interval, we associate speed with the model update interval (the other aspects of model update are the same as ECO). Adjusting the velocity-learning rate formula of the ALRCF, the following equation is obtained:

$$I = \left(1 - \frac{1}{1 + \left(\frac{6}{1+V}\right)^6}\right) \times I_{\max} \tag{8}$$

where V is the speed of the target, I is the template update interval, and $I_{\max}$ is the preset maximum update interval, updated every 8 frames. The dynamic update interval curve is shown in Fig. 2.

When the speed is low, the appearance of the target changes only slightly. The model's initial information should be maintained as much as possible, corresponding to a large update interval. When the speed is high, the appearance of the target changes considerably and should be updated to ensure the discriminative power of the model, corresponding to a smaller update interval. Algorithm 1 shows the procedure for the tracker.

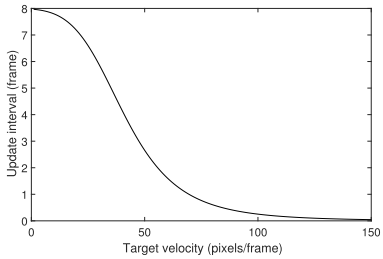

Fig. 2 Curve of velocity and learning rate.

Algorithm 1 Our tracker's algorithm

Input:
x: training image patch
P_{old}: previous frame position
S_{old}: previous target scale

Output:
P_{new}: new position
S_{new}: new target scale

1: **translation estimation**
- The response score is calculated according to Eq. (2).
- Set P_{new} to the target position that maximizes the response.

2: **Scale estimation**
- Calculate the prediction scale $S_{new'}$ according to S_{old} using Eq. (7).
- Extract scale sample with scale $S_{new'}$ at P_{new} and calculate the response score.
- Set S_{new} to the target scale that maximizes the response.

3: **Model update**
- Calculate the update interval using Eq. (8).
- Update the model when the update interval is reached.

3. Experiments

3.1 Experimental Setup

Our proposed approach (PSCT) was evaluated alongside four other state-of-the-art methods on the OTB2015 and KITTI datasets. KCF [8] and algorithms that use multi-scale strategies were selected, specifically SAMF [4], DSST [6], CCOT [5], and ECO-HC [14]. Twenty fast motion sequences were selected from the OTB2015 [3] and KITTI [13] datasets to validate the algorithms' ability to track fast motion scenes.

All algorithms were run with Matlab 2016b on an Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz with 8 GB RAM. Histogram of Gradient (HOG) and Color Names (CN) were used for feature representations. The filter size was four times the target size, and HOG features were extracted using a cell size of 4×4. We utilized S = 17 scales with a scale factor of a = 1.07. The initial value of the rate of change of the scale was $\mu_1 = 0.5$. To compare the performance of the different trackers, the algorithm was evaluated using spatial robustness evaluation (SRE) via success and precision plots. The success and precision plots illustrate the overlap precision (OP) and distance precision (DP) of the trackers over a range of thresholds. In the success plot, the trackers are ranked according to the area under the curve (AUC) score.

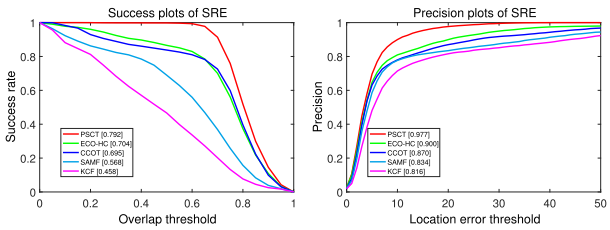

Fig. 3 Success and precision plots for the 20 fast motion video sequences selected from OTB2015 and KITTI datasets. Success plots use mean AUC for ranking and precision plots use threshold = 20 for ranking.

Table 1 Accuracy and speed comparisons on the test dataset. The best two results are shown in red and blue fonts, respectively.

	KCF	SAMF	CCOT	ECO-HC	PSCT
Success	0.458	0.568	0.695	0.704	0.792
Precision	0.816	0.834	0.870	0.900	0.977
FPS	203.9	18.4	5.3	54.2	54.9

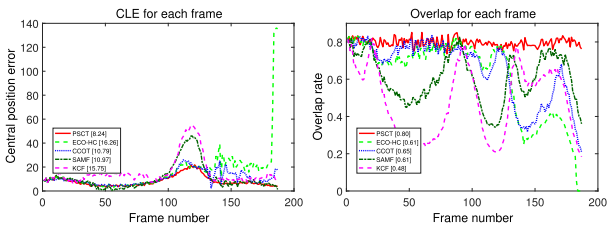

Fig. 4 Visualization of the position error and overlap rate for each frame. The results show that our tracker has better tracking capability than other state-of-the-art methods for fast motion scenes.

3.2 Performance Comparison

Figure 3 show the ranking scores for the precision and success plots. The AUC score of our tracker attained a value of 79.2%, which was 8.8%, 9.7%, 22.4%, and 33.4% higher than the ECO, CCOT, SAMF, and KCF algorithms, respectively. It can be seen from Fig. 3 that the mean DP score is 97.7%, which is greater than that of the algorithms. In Table 1, the accuracies and speeds of the trackers on the OTB2015 and KITTI datasets are summarized. Among these methods, the tracker achieves excellent performance.

A set of video sequences was selected for the purpose of analyzing the changes in the CLE and the overlap rate, and comparing the performance of the different algorithms. It can be seen from Fig. 4 that between frames 94 to 113, the SAMF and KCF algorithms cannot keep up with the scale change, hence why the overlap rate decreased from 77.4% to 21% and the location error increased from 10.7 pixels to 50.3 pixels. The overlap rate continues to fluctuate in subsequent frames. At roughly frame 160, the performance of the ECO tracker did not decrease rapidly, though the distance of the target increased. Thus, the overlap rate decreased from 39.8% and the location error began to fluctuate until frame 184, at which point the position error increased rapidly and the overlap rate dropped to zero. However, the overlap rate

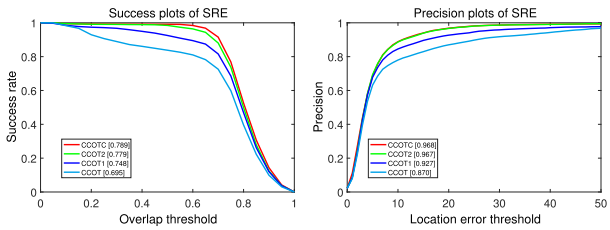

Fig. 5 Comparison of performance between the optimized algorithm (CCOTC) and the original algorithm (CCOT). CCOT, CCOT1, and CCOT2 correspond to the original algorithms with scales of 7, 9, and 11, respectively.

of our algorithm maintained a value of 80% despite repeated changes in the scale of the target, and the location error held steady at approximately 10 pixels.

To verify the generality of the optimization method presented in this paper, it was tested against the CCOT algorithm. The advanced algorithm (CCOTC) was compared to CCOT in 20 fast-moving target scenarios. CCOT1 and CCOT2 refer to the same algorithm with scale numbers of 9 and 11, respectively. In Fig. 5, it can be clearly seen that CCOTC performs better than the original algorithm.

4. Conclusion

In this paper, we proposed an improved multiscale tracking algorithm to solve the problem of keeping up with changes in the target's scale during fast motion. This was achieved using multiscale tracking based on the scale of the prediction rather than on the scale from the previous frame. Furthermore, feedback control was used to improve the prediction accuracy, and adaptive model updates based on speed further improved the model's robustness. Experiments on the OTB2015 and KITTI datasets demonstrated that our approach outperformed most existing state-of-the-art trackers in fast motion scenarios.

References

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

[2] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, et al., "The sixth visual object tracking VOT2018 challenge results," Proc. European Conference on Computer Vision (ECCV), pp.3–53, 2018.

[3] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no.9, pp.1834–1848, Sept. 2015.

[4] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," European Conference on Computer Vision, ECCV 2014, Lecture Notes in Computer Science, vol.8926, pp.254–265, Springer International Publishing, Cham, 2015.

[5] M. Danelljan, A. Robinson, F.S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," European Conference on Computer Vision, ECCV 2016, Lecture Notes in Computer Science, vol.9909, pp.472–488, Springer International Publishing, Cham, 2016.

[6] M. Danelljan, G. Häger, F.S. Khan, and M. Felsberg, "Discriminative scale space tracking," IEEE Ttrans. Pattern Anal. Mach. Intell., vol.39, no.8, pp.1561–1575, 2017.
[7] J. Pi, S. Zeng, Q. Zuo, and Y. Wei, "Accurate scale adaptive and real-time visual tracking with correlation filters," IEICE Trans. Inf. & Syst., vol.E101-D, no.11, pp.2855–2858, Nov. 2018.
[8] 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.
[9] M. Danelljan, G. Häger, F.S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," 2015 IEEE International Conference on Computer Vision, pp.4310–4318, 2015.
[10] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.489–497, 2018.
[11] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.4670–4679, 2019.
[12] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," European Conference on Computer Vision, ECCV 2012, Lecture Notes in Computer Science, vol.7575, pp.702–715, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
[13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," International Journal of Robotics Research, vol.32, no.11, pp.1231–1237, 2013.
[14] M. Danelljan, G. Bhat, F.S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," IEEE Conference on Computer Vision and Pattern Recognition, pp.6931–6939, 2017.
[15] C.S. Asha and A.V. Narasimhadhan, "Adaptive learning rate for visual tracking using correlation filters," Procedia Computer Science, vol.89, pp.614–622, 2016.