Correlation filter tracking based on superpixel and multifeature fusion*

ZHANG Hong-ying (张红颖)**, **WANG Hui-san** (王汇三), **and HE Peng-yi** (贺鹏艺)
College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China

(Received 22 November 2019; Revised 6 January 2020)

For the correlation filtering (CF) tracking algorithm is not robust enough and cannot adapt to scale changes, target occlusion (OCC) and other complex interferences. We introduce a CF tracking algorithm based on superpixel and multifeature fusion (CFSMF). First, superpixel segmentation and clustering are performed for the target and its surrounding environment in the initial frame. Then, a target appearance is reconstructed through block segmentation-based overlapping analysis to remove redundant information. On this basis, the histogram of gradient (HOG) and HSI color features of the target sub-block are extracted to interact with their respective position filters. Accordingly, the target position is determined by the weighted fusion of the response values. In the scale prediction stage, we independently train a scale filter with a multiscale pyramid constructed at the estimated target location. The object scale is estimated in terms of the filter response, thereby enabling the tracking algorithm to adapt to the object scale change. Lastly, we introduce an OCC criterion for determining whether to update the model or not. Compared with the classical tracking algorithm kernelized correlation filters (KCF), the proposed algorithm boosts the tracking success rate by 20% and tracking accuracy by 15.9%. Our algorithm in this paper could track the target stably even when the target is occluded and its scale changes.

Document code: A **Article ID:** 1673-1905(2021)01-0047-6
DOI https://doi.org/10.1007/s11801-021-9198-2

Object tracking has always been a major topic in computer vision research and is widely used in various fields[1]. The existing tracking algorithms can be divided into discriminative[2-4] and generative trackers[5-7] according to the manner by which surface models are built.

The correlation filtering (CF) tracking method[6-9] has gained extensive attention from numerous researchers due to its ability to convert the calculation from time to frequency domain for fast learning and detection. The initial CF algorithms only used grayscale features, such as MOSSE[10] and nuclear cycle structure tracker[11]. In Ref.[12], they proposed kernelized correlation filters (KCF), which used histogram of gradient (HOG) features instead of the original grayscale ones. This algorithm improves tracking accuracy and ensures the tracking speed in the meantime. However, the KCF algorithm has a single feature and cannot adapt to the object scale change. In Ref.[13] they used color name (CN) features in the CF framework and reduced the color features from 11D to 2D to ensure the tracking speed. In Ref.[14], an adaptive feature selection method was proposed by analyzing the tracking performance of different features. A suitable feature would be selected to predict the target position. In Ref.[15], a classifier combining a template with histogram features was proposed to train color and HOG features and merge the extracted ones.

Most tracking algorithms use high-level surface structural information, low-level clues, or both to represent and match targets. Accordingly, the research and application of mid-level visual clues during tracking are ignored.

The block-based tracking method[9-11] has recently received increasing attention due to its robustness for partial occlusion. Wang et al[16] proposed the superpixel tracking algorithm, which maximizes the superpixel characteristics and the background information around the target. Accordingly, the tracker has good robustness to deal with heavy occlusion and out-of-plane rotation. However, the performance under heterogeneous lighting conditions is unsatisfactory.

The DSST[17] treats visual tracking as two independent problems and applies separate discriminating CFs for translation and scale estimations. This method first uses the HOG feature to train a translation CF for detecting the translation of the target center. Then, a scale CF is trained to search the optimal scale on the multiscale spatial pyramid.

* This work has been supported in part by the National Key Research and Development Project of China (No.2018YFB1601200), the Key Projects of the Civil Aviation Joint Fund of the National Natural Science Foundation of China (No.U1533203), and the Fundamental Research Funds for the Central Universities (No.3122018C004).
** E-mail: carole_zhang@vip.163.com

LCT[18] predicts the scale through a complete search of the target's external pyramid and estimates the translation by modeling the temporal context correlation. SRDCF[19] makes use of a negative Gaussian penalty weight on the filter parameters to overcome the boundary effect. Deep SRDCF[20] introduces CNN features into SRDCF[19] and achieves good results. C-COT[21] further converts feature maps of different resolutions into a continuous spatial domain to achieve better accuracy. The subsequent ECO[22] improves the C-COT[21] tracker in terms of performance and efficiency.

In this study, we propose a CF algorithm based on a superpixel block and multi-feature to cope with the aforementioned complex scene, particularly to achieve real-time tracking with high accuracy and robustness. Fig.1 shows the framework of the tracking algorithm proposed in this article.

In this work, we use the simple linear iterative clustering algorithm (SLIC)[23], which is dull and computationally fast, for superpixel segmentation.

First, the initial target location and search area are defined as {center, size} and {center, search_size}, respectively, by using the first frame. The extremely large or small search area could adversely affect the completion of subsequent tracking tasks. On this basis, we set the search area size to 1.5 times that of the target size. The initial superpixel clustering center is assumed to be k, and the distance between adjacent superpixel is $S = \sqrt{N/k}$. Then, similar to the SLIC superpixel segmentation process, the k-means algorithm is used for iterative optimization in the pixel neighborhood until the error converges.

We calculate the gradient of each pixel in the clustering center neighborhood of $S\times S$ and move the center to the place with the minimum gradient. Afterward, we segmentation the superpixel labels in the neighborhood of $2S\times 2S$ and recalculate the distance from each pixel to the cluster center, as shown in Eq.(1):

$$\begin{cases} d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\ d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\ D = \sqrt{(d_c/N_c)^2 + (d_s/N_s)^2} \end{cases}, \tag{1}$$

where l, a and b are the color values of the pixel points, x and y are the ordinates of the pixels, N_c denotes the maximum distance of color space, and N_s is the maximum spatial distance within the class.

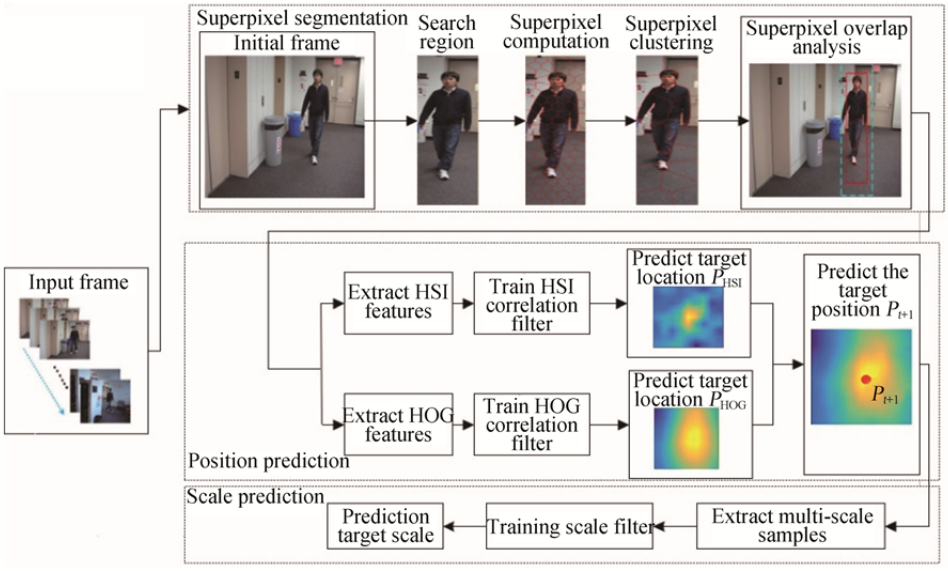

Fig.1 Framework of the CFSMF

Lastly, we solve the problem through iteration until the error converged. Then, we eliminate the small-sized superpixel through combination with the operation of enhanced connectivity. These steps were undertaken to complete superpixel segmentation. Fig.2 shows the effect of superpixel segmentation. The green bounding box represents the search region, and the yellow one is the target region.

We analyze the superpixel overlap degree after segmentation to remove redundant information further. In each superpixel block, we denote the pixels located with

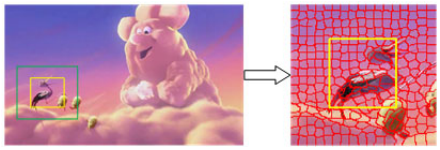

Fig.2 Diagram of superpixel segmentation

in the initial target range as the target pixels N_i^+; otherwise, they are denoted as background pixels N_i^-. Thus, the overlap degree O_i of each superpixel block in the search area is defined as

$$O_i = \frac{N_i^+}{N_i^+ + N_i^-} \ . \tag{2}$$

If the superpixel overlap value is less than 0.5, the superpixel block is the central part of the background rather than the target. We exclude this type of superpixel blocks and only keep the central part of the superpixel blocks belonging to the target mode.

Then we extract HSI color features and HOG features based on superpixel segmentation to represent the information of the target. The two features complement each other and jointly achieve the feature expression of the tracking target. In this work, we use the weighted adaptive method to achieve multi-feature fusion and input the two features into the CF for training as Eq.(3) and Eq.(4):

$$\hat{f}(z_{\mathrm{HSI}}) = k^{\hat{x}_{\mathrm{HSI}} z_{\mathrm{HSI}}} \ \hat{\alpha}_{\mathrm{HSI}} , \tag{3}$$

$$\hat{f}(z_{\mathrm{HOG}}) = k^{\hat{x}_{\mathrm{HOG}} z_{\mathrm{HOG}}} \ \hat{\alpha}_{\mathrm{HOG}} . \tag{4}$$

We can obtain position-related filters $\{\hat{x}_{\mathrm{HSI}}, \hat{\alpha}_{\mathrm{HSI}}\}$ and $\{\hat{x}_{\mathrm{HOG}}, \hat{\alpha}_{\mathrm{HOG}}\}$. The corresponding maximum output of the positive-dependent filter is the predicted position of the target and is shown as Eq.(5) and Eq.(6):

$$p_{\mathrm{HSI}} = \arg\max_{p_{\mathrm{HSI}}} F^{-1}\left(\hat{f}\left(z_{\mathrm{HSI}}\right)\right), \tag{5}$$

$$p_{\mathrm{HOG}} = \arg\max_{p_{\mathrm{HOG}}} F^{-1}\left(\hat{f}\left(z_{\mathrm{HOG}}\right)\right). \tag{6}$$

The distance and proximity of the training sample to the target location depend on the size of the response value of the associated filter. When the sample is close to the target, the filter response is high, the distance is great, and the response is small. In this case, the weighted coefficient of feature fusion is determined on the basis of the difference between the two features and the corresponding filter. We can express the position prediction weight factor δ as

$$\delta = \frac{\max\left(f\left(Z_{\mathrm{HSI}}\right)\right)}{\max(f(Z_{\mathrm{HSI}})) + \max(f(Z_{\mathrm{HOG}}))}, \tag{7}$$

where the range values of δ is [0,1]. Eq.(8) can obtain the final predicted position of the target:

$$p = \delta p_{\mathrm{HSI}} + (1-\delta) p_{\mathrm{HOG}} \ , \tag{8}$$

where p_{HSI} and p_{HOG} are the output values of the related filters for HSI and HOG features, respectively.

Because the target scale changes between two frames are often smaller than the target position changes during tracking, we predict the target position and then train a scale filter to estimate the target scale change based on the predicted target position. The specific steps are as follows.

In frame t with $M\times N$ target size, a scale pyramid image block x_{S} is created around the predicted target location by

$$x_{\mathrm{S}}=\alpha^n M\times\alpha^n N, \tag{9}$$

where α is the scale factor set to 1.02, and S is the size of the scale filter. The dimensions of the scale pyramid blocks with different sizes become uniform and are consistent with those of the scale filter by utilizing bilinear interpolation. We choose the HOG feature as the feature representation of x_{S} to guarantee the object feature shape. Similar to Eq.(2), we can obtain the coefficient matrix of scale filter $\hat{\boldsymbol{\alpha}}_{\mathrm{s}}$ through

$$\hat{\boldsymbol{\alpha}}_{\mathrm{s}} = \frac{\hat{\boldsymbol{y}}_{\mathrm{s}}}{\hat{\boldsymbol{k}}^{x_{\mathrm{s}}x_{\mathrm{s}}} + \lambda} \ . \tag{10}$$

After constructing the scale filter in frame t, we can predict the target scale in frame t+1. We extract the image blocks of scale pyramids with the target location predicted by frame t as the center. The HOG features are extracted to form the sample set z_{S} to be tested. The formula is shown as

$$\hat{f}_{\mathrm{s}}(z_{\mathrm{s}}) = \hat{\boldsymbol{k}}^{x_{\mathrm{s}}z_{\mathrm{s}}} \ \hat{\boldsymbol{\alpha}}_{\mathrm{s}} \ , \tag{11}$$

where $f_{\mathrm{S}}(z_{\mathrm{S}})$ is the regression function of the scale filter. The scale value corresponding to the maximum value of $f_{\mathrm{S}}(z_{\mathrm{S}})$ is the scale transformation of the frame. The scale filter parameters are updated via linear interpolation.

The learning rate should be dynamically adjusted in accordance with the target changes. We set a small learning rate at this time to ensure tracking stability. When the target dramatically changes, an extensive learning rate should be set to update the model rapidly. In this work, we measure the target change of two adjacent frames by calculating the average difference of two adjacent frames. In the $M\times N$ image, the pixel size is denoted by I_{ij}. The average difference between the images of frames t and $t-1$ can be obtained by

$$d = \frac{\sum_{i,j}^{M,N} |\,(I_{ij}^{t} - I_{ij}^{t-1})\,|}{MN} \ . \tag{12}$$

When d<3, we set a low learning rate η=0.025. When 3≤d<8, we set a moderate learning rate η=0.05. When d≥8, we set a high learning rate η=0.1. This adaptive piece-wise learning rate method makes the improved algorithm robust in tracking complex scenarios.

We introduce an OCC detection mechanism to determine whether the model is updated or not by judging the degree of oscillation of the output response value. This task is performed to improve the anti-occlusion ability of the algorithm and prevent the tracking drift when the model interferes with the discrimination ability of the model by learning substantial background information. The OCC detection mechanism is as

$$\omega = \frac{\left|F_{\max} - F_{\min}\right|^2}{\mathrm{mean}\left(\sum\left(F_{m,n} - F_{\min}\right)^2\right)} \ , \tag{13}$$

where $F_{\max}$ represents the corresponding maximum value of the CF output, $F_{\min}$ represents the corresponding minimum value of the CF output, and $F_{m,n}$ represents the value of any position in the correlation output confidence graph. If the value of ω suddenly decreases, the tracking target has OCC, and the model is not updated at this time.

The public dataset OTB-2015[24] exhibits 100 sequences with 11 attribute labels, including illumination

variation (IV), OCC, DEF, SV, and background clutter (BC). In our experiment, we use OTB-2015 for performance evaluation. The computer hardware used in the experiment is configured with Intel Core i7-6700HQ CPU (2.6 GHz) and 16 GB of memory. The software platform is MATLAB 2017a. Subsequently, we select nine representative and outstanding CF-based algorithms for comparison. These algorithms are listed as follows: KCF[12], DSST[17], LCT[18], SRDCF[25], DeepSRDCF[26], SAMF[19], HDT[20], Staple[27] and CF2[28]. The original parameter settings of the algorithms are retained to ensure fairness.

We use the precision and success plots to evaluate the above-mentioned tracking algorithms. Fig.3 illustrates the comparison results of OPE, SRE, and TRE over all the 100 sequences of OTB-100. Fig.3 shows that the proposed method in this study performs efficiently.

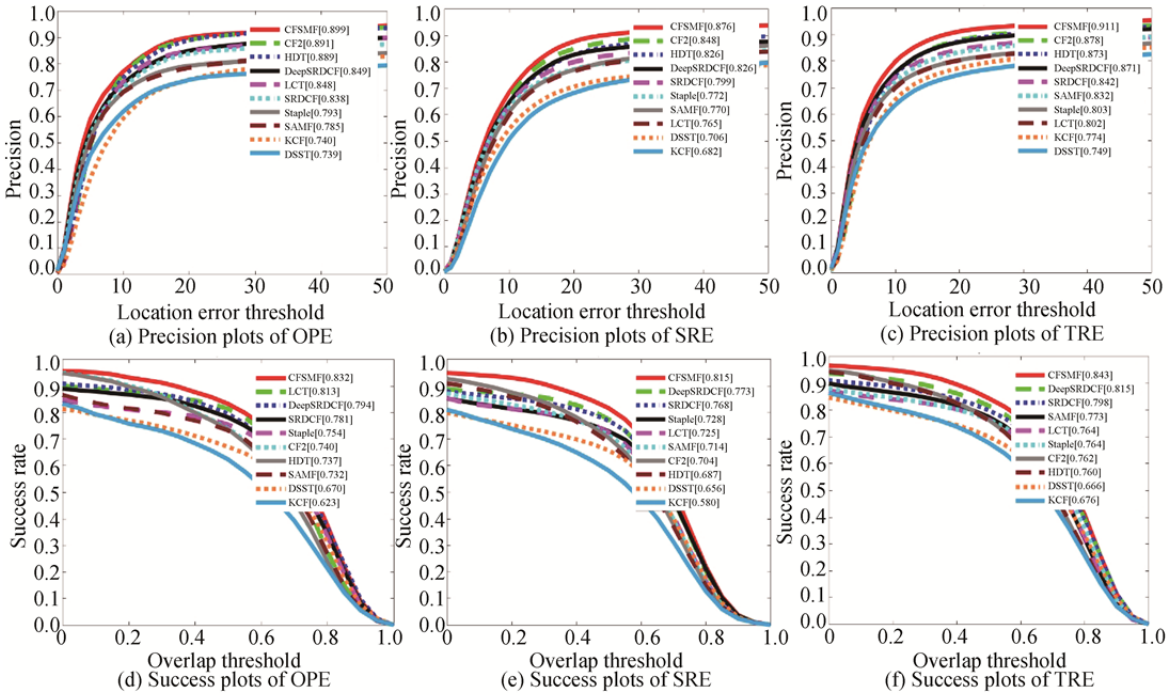

Fig.3 Precision and success plots of OPE, SRE and TRE for 10 trackers on the OTB-2015 benchmark

It could be seen that the CFSMF outperforms all the trackers in OPE, SRE and TRE. The OPE results show the one-pass evaluation. It can be seen that our approach ranks first in precision and success rates. In the OPE success plot, the AUC of our approach is 0.832, which is higher than that of the KCF approach by 20.09%. In the OPE precision plot, our tracker gains a precision score of 0.899, which exceeds the KCF by 15.9%.

In addition to the precision and success rates, the tracking speed is also an important evaluation index. The tracker's speed reflects whether the tracker can be used in real-time tracking. Tab.1 shows the comparison result in terms of average frame per second (fps). We run all the algorithms in the table under our experimental environment to ensure fairness. The table demonstrates that CFSMF achieves a speed of 54 fps, which ranks second among all 10 algorithms and the upper and lower speed bounds of CFSMF are 74 fps and 31 fps, respectively. The tracker requires a running speed of ≥25 fps to achieve real-time tracking. The table manifests that CFSMF is suitable for real-time tracking.

Tab.1 Speed comparison of tracking algorithms on the OTB-2015 dataset

Tracker	HDT	LCT	DSST	SAMF	Staple	SRDCF	Deep SRDCF	KCF	CF2	CFSMF
Ave. (fps)	10	25	41	17	65	3	1	195	11	54

Fig.4 shows attribute analytical plots of BC, IV, OCC, and SV. The results of these annotated sequences can help evaluate the advantages and weaknesses of the tracking methods. The performance of our tracking approach under different challenging situations is experimentally tested. Fig.4 shows the OPE precision and success plots. CFSMF shows optimal performance on all four attributes and achieves higher rates compared with the baseline KCF tracker on each attribute. Fig.4 presents that our approach optimally performs in precision and success plots. This

finding suggests that the CFSMF can effectively cope with IV, OCC and SV situations. In the BC attribute precision plots, CFSMF achieves the third-best performance, which is close to the best performance CF2.

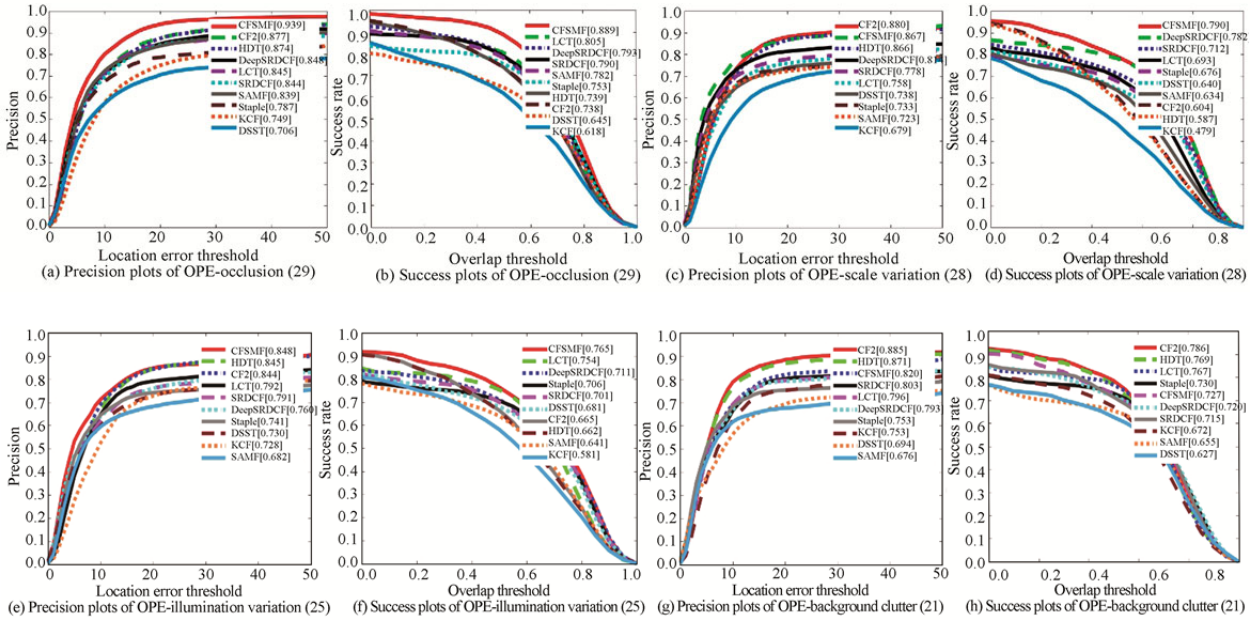

Fig.4 Attribute analysis with precision and success plots

The accuracy and robustness of the algorithm can be intuitively demonstrated through qualitative analysis. In Fig.5, we select four representative and good performance approaches, namely, KCF, SRDCF, Staple, and Struck, to test our method. The "Freeman4" sequence tracks the head of a person who freely walks in a cluttered classroom. The target is small and goes through various complex scales and similar occlusion. The graph shows that KCF, Staple, and CFSMF can steadily track. However, CFSMF is optimal in terms of tracking accuracy. The SRDCF and Struck have lost the object. In the "CarScale" sequence, the KCF and Struck accuracies are insufficiently high because they cannot adapt to scale changes even though all algorithms have successfully tracked the target. In the "Couple" sequences, the scales and appearances of the object change during the tracking process. Although several trackers can accurately locate the targets throughout the entire sequence, the bounding box sizes of KCF and Struck cannot change as much as the ground truth in this sequence. By contrast, our approach can sufficiently change the size of its bounding box size during tracking. In the "Car2" sequences, all trackers, including CFSMF, can accurately track the targets throughout the entire sequence.

In view of the problems of tracking failure due to complex situations, such as target occlusion and scale change, a new method for modeling target appearance is proposed. Object appearance is reconstructed by using superpixel segmentation and clustering, which improve computational efficiency and have strong adaptability.

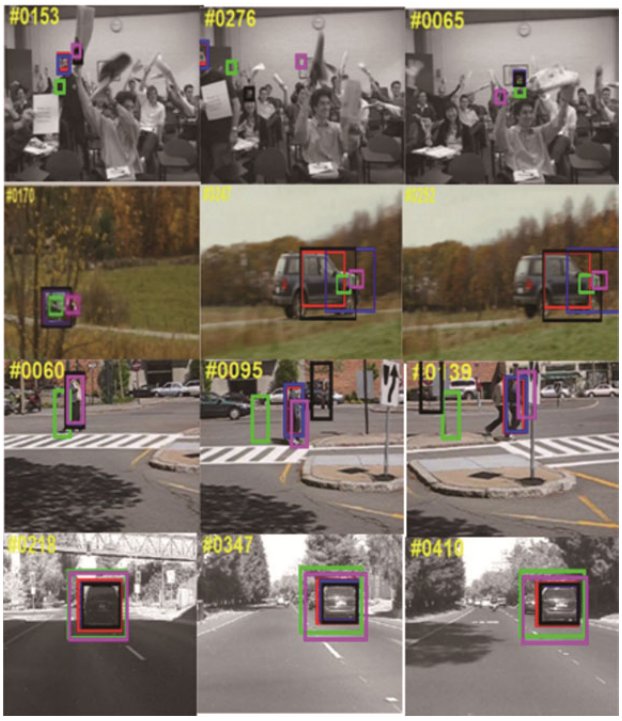

Fig.5 Tracking results of the dataset OTB-2015 (From top to bottom: Freeman4, CarScale, Couple, Car2)

The HOG and HSI color features are then extracted in accordance with their complementary features and fused to characterize the target. These approaches make the

algorithm robust when coping with light changes and target deformation. Target position prediction is completed under the framework of the previous CF, and the tracking accuracy of the position filter is improved. An independent scale-dependent filter is used to achieve the adaptation of the target scale change. The final combination of the response values of the position and scale filters completes the target tracking.

References

[1] D. A. Ross, J. Lim, R. -S. Lin and M H. Yang, International Journal of Computer Vision **11**, 125 (2008).

[2] Jia Xu, H. Lu and M. H. Yang. Visual Tracking via Adaptive Structural Local Sparse Appearance Model, in Proc. IEEE Conf. Computer. Vis. Pattern Recognition, 1822 (2012).

[3] N Y Wang and D Y Yeung, Learning a Deep Compact Image Representation for Visual Tracking, International Conference on Neural Information Processing Systems Curran Associates Inc., 809 (2013).

[4] Adam Amit, E. Rivlin and I. Shimshoni, Robust Fragments-based Tracking using the Integral Histogram, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 798 (2006).

[5] J Shen and S Zafeiriouand GChrysos, The First Facial Landmark Tracking in-the-Wild Challenge: Benchmark and Results, IEEE International Conference on Computer Vision Workshops, 1003 (2015).

[6] M Matthias, N. Smith and B. Ghanem, Context-Aware Correlation Filter Tracking, IEEE Conference on Computer Vision and Pattern Recognition, 1387 (2017).

[7] H Sheng, K Lv, J H Chen and W li, Robust Visual Tracking Using Correlation Response Map, IEEE International Conference on Image Processing, 2381 (2016).

[8] Li Y and Zhu J, A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration, European Conference on Computer Vision, Springer, Cham, 2014.

[9] Liu J M, Guo J W and Shi S, Optics and Precision Engineering **26**, 2100 (2018).

[10] Bolme D S, Beveridge J R, Draper B A and Y M Lui, Visual Object Tracking Using Adaptive Correlation Filters, IEEE Conference on Computer Vision and Pattern Recognition, 2544 (2010).

[11] J. F. Henriques, R. Caseiro, P. Martins and J. Batisita, Exploiting the Circulant Structure of Tracking-by-detection with Kernels, the 12th European Conference on Computer Vision, 702 (2012).

[12] J. F. Henriques, R Caseiro, P Martins and J Batisita, PAMI **37**, 583 (2015).

[13] D. Martin, F. Khan, M. Felsberg and J. Weijer, Adaptive Color Attributes for Real-Time Visual Tracking, IEEE Conference on Computer Vision and Pattern Recognition, 1090 (2014).

[14] B. Luca, J. Valmadre, S. Golodetz, O. Miksik and P. H. Torr, Staple: Complementary Learners for Real-Time Tracking, IEEE Conference on Computer Vision and Pattern Recognition, 1401 (2016).

[15] X H Yang, H Zhang, L Yang, C S Liu and X. Peter, A Joint Multi-Feature and Scale-Adaptive Correlation Filter Tracker, IEEE Access **PP**, 1 (2018).

[16] S. Wang, H. Lu, F. Yang and M.-H. Yang, Superpixel Tracking, IEEE International Conference on Computer Vision, Barcelona, 1323 (2011).

[17] Danelljan M, Häger G, Khan F S and Felsberg M, IEEE Transactions on Pattern Analysis & Machine Intelligence **39**, 1561 (2016).

[18] C Ma, X K Yang, C Y Zhang and M S Yang, Long-term Correlation Tracking, IEEE Conference on Computer Vision and Pattern Recognition, 5388 (2015).

[19] Li Y and Zhu J, A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration, European Conference on Computer Vision, 254 (2016).

[20] Y K Qi, S P Zhang, L Qin, H Yao, Q Huang, J Lim and M H Yang, Hedged Deep Tracking, IEEE Conference on Computer Vision and Pattern Recognition, 4303 (2016).

[21] 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, 2016.

[22] M. Danelljan, G. Bhat, F. S. Khan and M Felsberg, ECO: Efficient Convolution Operators for Tracking, IEEE Conference on Computer Vision and Pattern Recognition, 21 (2017).

[23] R Achanta, A Shaji, K Smith, A Lucchi, P Fua and S Susstrunk, IEEE Transactions on Pattern Analysis and Machine Intelligence **34**, 2274 (2012).

[24] Wu Yi, J. Lim and M. H. Yang, IEEE Transactions on Pattern Analysis and Machine Intelligence **37**, 1834 (2015).

[25] M. Danelljan, G. Hager, F. Shahbaz Khan and M. Felsberg, Learning Spatially Regularized Correlation Filters for Visual Tracking, International Conference on Computer Vision, 4310 (2015).

[26] Danelljan M, Hager G, Khan F and M Felsberg, Convolutional Features for Correlation Filter Based Visual Tracking, IEEE International Conference on Computer Vision Workshop, 621 (2015).

[27] L. Bertinetto, J Valmadre, S Golodetz, O Miksik and P Torr, Staple: Complementary Learners for Real-Time Tracking, IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[28] C Ma, J Huang, X Yang and M Yang, Hierarchical Convolutional Features for Visual Tracking, International Conference on Computer Vision, 2015.