Visual Tracking based on Cooperative model

Zhang Bobin[1], Fang Weidong[2], Chen Wei[1,2 *],
[1] School of Computer Science and Technology
China University of Mining and Technology
Xuzhou, Jiangsu, 221116, China
[2] Key Laboratory of Wireless Sensor Network and Communication
Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences
Shanghai 201899, China

Bi Fangming[1],Tang Chaogang[1], Huang Xiaohua[3]
[3] Center for Machine Vision and Signal Analysis (CMVS) & Faulty of Information Technology and Electrical Engineering (ITEE)
University of Oulu
Oulu, FI-90540, Finland

[*] Corresponding author: Chen Wei
chenw@cumt.edu.cn

***Abstract*—In this paper, we propose a cooperative model combined the multi-task reverse sparse representation model (MTRSR) and the AdaBoost classifier, which were used to cope with the disturbing of target gradient information caused by motion blur or target serious occlusion, and a descriptive dictionary were used to estimate the weights of each candidates. First, we use the MTRSR model to get the blur kernel which were used to get the blur target template set, meanwhile the confidence of the candidates is also obtained by the reconstruction error. Then we use the HOG features of the target templates to get the descriptive dictionary to calculate the weights of the candidates, and a AdaBoost classifier is used to calculate the confidences of all candidates. Finally, the best target is retrieved by the sum of production of weight value and the two confidences. The experimental data show that the proposed algorithm can fully cope with the target's information change which were caused by motion blur and target occlusion in the complex scene, and our algorithm can further improve the accuracy and robustness in visual tracking.**

Keywords—Visual tracking; AdaBoost classifier; sparse representation; collaborative model

I. INTRODUCTION

Visual tracking plays an important role in computer vision and image processing, since it has been widely applied to vision navigation, intelligent transportation and video surveillance. In recent years, more research have been obtained such as: [1],[2],[3],[4],[5],[6],[7],[8],[9],[10]. But the visual tracking still faces many challenges: 1) The videos sometimes introduce the motion blur which can change the structure information of the target area with pixel intensity and the gradient that makes it impossible to accurately identify the best candidates, and leads to the tracking drift or losing. 2) When the target was seriously occluded or the occlusion is similar with the target. The algorithm may keep the occlusion as the best target which cause the target losing.

Therefore, Bao et al. proposed an Accelerated Proximal Gradient L1 tracking algorithm (L1APG)[11], which can effectively and quickly solve the minimization problem of L1 normal constraints and ensure quadratic convergence of its solution. However, the algorithm cannot effectively cope with the serious occlusion of candidates and the change of gradient information caused by motion blur in video sequences, therefore, the tracking results are sometimes not stable when the information of the target area is changed. Zhang et al. proposed a Continuous Low Rank Sparse Tracking algorithm (CLRST) [12], which uses time consistency and adaptively select candidate particles. To some extent, the algorithm can deal with the deformation and partial occlusion problem of target more robust. However, when the target and background are very similar and the motion blur or serious target occlusion occurred, especially when the occlusion and the target have similar appearance, it will produce very similar target information. These situations cause the algorithm cope with it ineffectively. Ma, B., et al. proposed a Multi-task Reverse Sparse Representation model (MTRSR)[13], which combines the estimation of blur kernel and the sparse representation of targets in a joint framework. To avoid introducing noise and ringing effects in the process of deblurring, the blur kernel is not used to restore targets, but convoluted with the clear templates to get the blur target template set. Specifically, the blur target template set is sparsely matched with the candidates, and then the sparse coding matrix C is obtained. The number of candidates is more than the target template set T, so the obtained C can eliminate the candidates that are not related to the target templates which can reduce the computational cost when matching the target. It is the first time to combine the blur kernel estimation and sparse representation model in a multi-task manner to deal with the problem of motion blur in video sequences. The combination strategy can get a single blur kernel k and sparse matrix C which can effectively and quickly eliminate candidates that are not related to the targets by iterative optimization. But when the object is seriously occluded and the edge gradient changes dramatically which keep it difficult to track targets effectively and robustness will lead the tracking drifting or target losing. Therefore, this paper proposes a visual tracking algorithm based on descriptive dictionary combined generative and discriminative method for handling the change of gradient information and object occlusion in target area.

The main contributions of this paper are as follows:

978-1-5386-2335-0/18/$31.00 2018 IEEE

In this paper, we use the method considering the generative and discriminative method to track the target which is more robustness and accurate in cope with the change of the target information and the serious target occlusion in video sequence. We simultaneously use two dictionaries for visual tracking, the one of which was set up by extract the vectored local patches from the target area is named as D_1. In our method, we use it to matching with the candidates for getting the sparse coding coefficients to training the AdaBoost classifier, while. the another is a descriptive dictionary, namely, D_2, which was obtained from the HOG features of the target templates that can preferably determine the weights of the candidates based on the information of the appearance gradient in target area.

By solving the HOG features of the target templates, we can obtain a descriptive dictionary D_2, and calculate the HOG features Y^H of candidates Y. Meanwhile, the weights of the candidates are obtained according to the reconstruction errors of the candidates Y^H and dictionary D_2.

By solving the MTRSR model, we can get the blur kernel k and the blur target template set T^* to calculate the reconstruction error which were used to get the confidences of each candidate. The AdaBoost classifier is trained by the extracting positive and negative samples according the tracked targets. We can get the best target by the sum of production of weight and two confidences.

II. Proposed Tracking Algorithm

Firstly, we can get the tracking result as the initial target template set T_i (i=1, 2 ,.., m), where m = 8 represents the number of the target templates, by the method of real-time compressive tracking[14] which size is 32×32 pixel.

A. Solution of the Blur Kernel k

Ma,B. et al. proposed a multi-task reversal sparse representation model (MTRSR)[13] to solve the blur kernel k and sparse representation of the target by:

$$\left[\hat{k}, \hat{C}\right] = \arg\min_{k,C} \left\| k * T - YC \right\|_F^2 + \nu \left\| k \right\|_2^2 + \lambda \left\| C \right\|_{2,1}, \qquad (1)$$

where k is the blur kernel, Y denotes the candidates, and T means the target template set, *represents the convolution operator, C is the sparse coding matrix. As this model includes two variables, it can be transformed into two optimal solutions of the sub problem. The sparse coding matrix C is initialized by:

$$\hat{C} = \arg\min_{C} \left\| T - YC \right\|_F^2 + \lambda \left\| C \right\|_{2,1}, \qquad (2)$$

where C can be solved by Accelerated Proximal Gradient method[15].

The solution of k in sub problem 1: Fix C to solve the blur kernel k:

$$\hat{k} = \arg\min_{k} \left\| k * T - Y\hat{C} \right\|_F^2 + \nu \left\| k \right\|_2^2. \qquad (3)$$

This problem can be regarded as the least square problem with Tikhonov regularization, and it's closed-form solution is[16]:

$$\hat{k} = F^{-1}\left(\frac{\bar{F}(T) \otimes F(Y\hat{C})}{\bar{F}(T) \otimes F(T) + \nu I} \right), \qquad (4)$$

where F(.) represents the Fast Fourier Transform, $F^{-1}(\cdot)$ denotes inverse Fast Fourier Transform, $\bar{F}(\cdot)$ denotes the complex conjugate of F(.), $\otimes$ denotes element-wise multiplication, and I is an identity matrix.

The solution of C in sub problem 2**:** Given the blur kernel k, the sparse matrix C is obtained by:

$$\hat{C} = \arg\min_{C} \left\| \hat{k} * T - YC \right\|_F^2 + \lambda \left\| C \right\|_{2,1}, \qquad (5)$$

where C can be solved by Accelerated Proximal Gradient method[15].

Algorithm 1[13]: Solve the blur kernel k and sparse matrix C

Input: target template set T, candidates Y, parameter ν and λ
Output: blur kernel k and sparse coding matrix C

```
Initialize sparse coding matrix C by (2)
For  i= 1,2, …, n  do
        Solve blur kernel k by (4)
        Solve sparse coding matrix C by (5)
end
```

B. Design and Training of AdaBoost Classifier

In the first eight frames, 9 positive samples are obtained by pixel perturbation which sampling near the target that were tracked in each frame, and 150 negative samples are obtained through pixel perturbation in the eighth frame (each patch is resized in 32×32 pixels). These positive and negative examples are extracted by 16×16 patches with 8 pixels as the step length, and the all local patches are vectored. So we can get each training sample $X=\{x_i \mid i=1,2, \ldots,n\} \in R^{d\times n}$, and each of the x_i is a vectored local patch, where n represents the number of local patches. To get the training dictionary $D_1=\{d_1,d_2,\ldots,d_{n\times m}\} (\in R^{d\times(n\times m)})$, we extract local patches from

the target template set T (=$\{T_1,T_2,\ldots,T_m\}$) by the same method. Therefore, the sub patch x_i of each training sample X is encoded by dictionary D_1:

$$\min_{\alpha_i}\left\|x_i - D_1\alpha_i\right\|_2^2 + \lambda_2\left\|\alpha_i\right\|_1, \qquad (6)$$

where $\alpha_i \in R^{(n\times m)\times 1}$ is the sparse coding coefficient for training the classifier. As we extract n sub patches from training sample X, and select k sparse coding coefficient with its sub patches to training the classifier, C_n^* weak classifier can be trained by different sub patches, and the best classifier is selected by the minimum classification error. In our implementation, we train 60 weak classifiers, and then we select 45 of them as the final strong classifier $H(x)$ (each weak classifier is a naive Bias classifier).

C. The Selection of the Best Targets and Calculation of the Weights about the Candidates

We use the algorithm which combined the generative and discriminative method in tracking, and the blur kernel k were used to convolved with the target template set T to get the blur target template set T^*. The dictionary D_1^* can be extracted by the same method like the construction of dictionary D_1, and each candidate Y_i can extract local patches as $Y_i = \{y_k \mid k=1,2,\cdots,n\}$. The patch y_k is coded with D_1 and D_1^* by:

$$\begin{aligned}&\min_{\xi_k}\left\|y_k - D_1\xi_k\right\|_2^2 + \lambda_3\left\|\xi_k\right\|_1,\\ &\min_{\xi_k^*}\left\|y_k - D_1^*\xi_k^*\right\|_2^2 + \lambda_3\left\|\xi_k^*\right\|_1,\end{aligned} \qquad (7)$$

where $\xi_k^* \in R^{n\times m\times 1}$ is the sparse coding coefficient of the local patch y_k with dictionary D_1^*, and $\xi_k \in R^{n\times m\times 1}$ is the sparse coding coefficient of the local patch y_k with dictionary D_1. The reconstruction error can get by:

$$\eta_k^* = \|y_k - D_1^*\xi_k^*\|_2^2, \qquad \eta_k = \|y_k - D_1\xi_k\|_2^2 \qquad (8)$$

where η_k^* and η_k is the reconstruction error about the local patch $y_k \in R^{d\times 1}$ of the candidate Y_i with dictionary D_1^* and D_1. Therefore, the confidence of the candidate Y_i is:

$$P_i = \sum_{k=1}^{n}\left(\exp\left(-5\eta_k^*\right) + \exp\left(-5\eta_k\right)\right). \qquad (9)$$

On the other hand, we solve the HOG features of the target template set T to get the descriptive dictionary $D_2 \in R^{e\times m}$. In detail, for the candidate Y_i, i=1,2,…,N., the corresponding HOG feature $Y_i^H \in R^{e\times 1}$ is obtained. The HOG feature of the candidate Y_i is coded by dictionary D_2 as followed:

$$\min_{\beta_i}\left\|Y_i^H - D_2\beta_i\right\|_2^2 + \lambda_4\left\|\beta_i\right\|_1, \qquad (10)$$

where $\beta_i \in R^{m\times 1}$ denotes the sparse coding coefficient of candidate Y_i^H with dictionary D_2. Therefore, the reconstruction error of the candidate Y_i^H with dictionary D_2 is formulated as followed:

$$\varepsilon_i = \left\|Y_i^H - D_2\beta_i\right\|_2^2. \qquad (11)$$

According to formula 11, the weight of the candidate Y_i is calculated by using the followed method:

$$W_i = \exp(-5\varepsilon_i). \qquad (12)$$

Therefore, the best candidate is calculated by:

$$Y_j = \max_j W_j H(Y_j) + \theta W_j P_j \qquad (13)$$

D. Update Strategy of Template Set and the Classifier

Templates update: In order to enable the algorithm to effectively cope with the changes in the target pose and scene, we need to dynamic updating the target template set T, and at the same time, we collect the positive and negative samples in the current frame as training data set which were used to training the classifier to reduce the classification error. The error of the selected candidate Y_j and the target template set T is defined as followed:

$$\delta = \frac{1}{m}\sum_{i=1}^{m}\left\|Y_j - T_i\right\|_2^2. \qquad (14)$$

When $\delta<\delta_0$, it indicates that the tracked target area is not highly polluted. So we can use the candidate Y_j to update the target template set T by Occlusion-Aware Update strategy [17].

Classifier update: when $\delta<\delta_0$, we update the classifier. According to the target location we tracked, we collect the positive samples (9 samples per frame) by pixel perturbation, and update the negative samples every 5 frames (150 samples at the last frame) to train the classifier.

The pseudocode of the algorithm are as follows:

Algorithm 2：The proposed tracking method

Input： The tracking result o_1, o_2, …, o_m in the front m frames achieved by real-time compressive tracking algorithm[14] which were used to get the blur target template set T; the number of templates m; update frequency Φ;

Output： tracking results s_t, t= m+1, m+2, …, M

Initialization of the classifier：

1：72 positive samples N_p from the first m frames (9

samples per frame), 150 negative samples N_q form the mth frame; target set ψ=0.

2：extract the local descriptors from the samples [N_p, N_q].

3：Training the strong classifier $H(x)$ by the local descriptors.

4: While t= m+1, …,M do

5: get the candidates $Y = [Y_1, Y_2, \ldots, Y_N]$.

6: get the blur kernel k by Algorithm 1.

7: get the blur target template set T^* by the blur kernel k which were used to convolved with target template set T.

8: calculate the reconstruction error of the candidate Y_i with dictionary D_1 and D_1^* by the (8).

9: get the confidences of each candidates by (9).

10: extract the local descriptors from the sparse coding coefficients of the candidates Y, and use it to calculate the classification values of each candidate.

11: get the weights W_i for each candidates by (12).

12: get the best target s_t by the (13).

13: If the error is less than the predefined threshold ($\delta<\delta_0$).

14: use the occlusion-aware update strategy [17] to update the template set T.

15: extract 9 positive samples from the tracked target $\Rightarrow N_p$.

16: update the target set $\psi = [\psi, s_t]$.

17: if size(ψ)== Φ

18: update the target set ψ = 0.

19: get 150 negative samples $\Rightarrow N_q$

20: extract the local descriptors from the sparse coding coefficients of the samples [N_p, N_q].

21: retrain the classifier $H(X)$.

22: end if

23: end while

III. Experiments and Performance

Our algorithm maintains 8 templates in the tracking process, and 800 candidates are collected in each frame which means the number of the particles is 800. All target template set, training samples and candidates are 32×32 pixels. We choose 8 pixels as a step length and select 9 overlapped local blocks with 16×16 pixels in the target area, and use these local sparse coding coefficients to form descriptors. To construct the descriptors, we select 3 from 9 local sparse coding coefficients to perform connect operation which were used to training the classifier. The parameters are fixed as: $v=\lambda=\lambda_2=\lambda_3=\lambda_4$=0.01, θ=0.1, δ_0=0.5, k=3, n=40. To evaluate the performance of the algorithm, our algorithm compared with 6 kinds of representative algorithms: Motion Blur Tracking (MBT)[13], Accelerated Proximal Gradient L1 tracker (L1APG)[11], Least Soft-Threshold Squares Tracking (LSST)[18], Fast Compression Tracking (FCT)[19], Strong Classifier Tracking (SCT)[17], Consistent Low-Rank Sparse Tracking (CLRST)[12].

Table 1. 6 video sequence features

Sequence name	Characteristics of the sequence
Walking2	Scale Variation，Occlusion，Low Resolution
BlurCar4	Motion Blur，Fast Motion
BlurCar2	Scale Variation，Motion Blur，Fast Motion
FaceOcc2	Illumination Variation，Occlusion，In-Plane Rotation，Out-of-Plane Rotation
Subway	Scale Variation，Occlusion，Low Resolution
Trans	Illumination Variation, Scale Variation , Occlusion, Deformation

In order to ensure the reliability of the experimental results, the codes of the above-mentioned algorithms are provided by their authors, and all the parameters of the algorithms also use the initial value. The video used in the experiment is taken from OTB-100[20].

A. Qualitative Analysis

Figure 1 shows the partial tracking results of seven tracking algorithms on six public videos (named Walking2, BlurCar4, BlurCar2, FaceOcc2, Subway, Trans). The red box in the figure shows the tracking results of our algorithm. Comparing with other visual tracking algorithms, it is found that the compared algorithms failed to detect the target in the videos. The results shows that the proposed algorithm can well cope with the motion blur and severe target occlusion in video sequences. Motion blur is the most important factor which affects the video quality in the sequences of BlurCar2 and BlurCar4. Our algorithm and L1-APG have more stable tracking results against motion blur. The visual tracking results of algorithm CLRST on BlurCar4 were accurate, but it loses the target in the seventy-ninth frame in BlurCar2, and the other algorithms also appear the phenomenon that losing the target in two videos. It shows that our algorithm is effective in dealing with the problems of motion blur and fast moving of targets in video sequences.

Some challenging problems such as deformation, occlusion and low resolution occurred in Walking2 and Subway. It is found that our algorithm works better than other algorithms in terms of average overlap rate or average center location error.

In FaceOcc2, the factors which influence the quality of the video are target occlusion, in-plane rotation and out-of-plane rotation, our algorithm can also well cope with it. In the video sequence Trans, there are serious factors influence the video quality such as: deformation, occlusion, scale variation and illumination variation. Under the influence of many comprehensive factors, our algorithm has some drift and deviation.

Through the above 7 algorithms in 6 video sequences shows that our algorithm can effectively deal with the problem

of motion blur, scale variation and occlusion in video sequence. And compared with other algorithms our algorithm get the best tracking result in different application scenarios. The tracking results are as Fig.1.

Fig1. Visual tracking result

B. Quantitative Analysis

We use the average center location error and average overlap rate to evaluate the performance of each algorithm. If the average center location error of the algorithm is small, and the higher average overlap rate means that the performance of the algorithm is better, and the tracking result is more precise and reliable, (Our algorithm's average overlap rate is 12.2% higher than the second L1-APG).

The center location error can be calculated by:

$$center\quad error = \sqrt{(x - x_0)^2 + (y - y_0)^2} \qquad (15)$$

where (x, y) is the tracked center location coordinate, and (x_0, y_0) is the labeled center location coordinate.

The overlap rate can be calculated by:

$$overlap = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)} \qquad (16)$$

where R_G means the manual marks of the tracking results, and R_T means the tracking results obtained by each algorithms. The greater the overlap rate is, the closer the tracking results are to the real results, and Table 2 gives the average overlap rate of the 7 algorithms. In Table 3, the average center location error of the 7 algorithms is given. And the smaller the value is, the more precise the algorithm is. The algorithm reduces the average center location error of 26 pixels by second algorithm L1-APG.

Table 2 and table 3 show that our algorithm gets the best performance in sequences Walking2, Subway and BlurCar2. Our algorithm has the largest average overlap rate and the average center location error is the smallest. The average overlap rate of our algorithm is 60.4, and it is superior to the second algorithm L1-APG which is 48.2. The average center location error of our algorithm is 23.7, which is better than 49.7 of second algorithm L1-APG. The experimental results

show that our algorithm can achieve a more stable tracking results and get high robustness to complex scenes.

Table 2 Average overlap rate (%)

Video	MBT	L1APG	LSST	FCT	SCT	CLRST	Ours
Walking2	50.9	72.3	34.9	28.6	4.8	36.0	**75.0**
BlurCar4	15.9	**71.5**	53.4	4.9	33.1	69.1	52.1
BlurCar2	11.0	57.3	9.7	9.5	31.3	12.0	**59.5**
FaceOcc2	35.7	22.7	41.5	65.2	24.7	**74.0**	67.8
Subway	15.0	15.5	18.7	62.1	54.2	18.1	**64.5**
Trans	33.9	**50.0**	36.7	51.0	29.1	49.4	43.7
Average	27.1	48.2	32.5	36.9	29.5	43.1	**60.4**

Table 3 Average center location error (unit: pixel).

Video	MBT	L1APG	LSST	FCT	SCT	CLRST	Ours
Walking2	12.7	4.1	39.7	59.3	102.6	38.9	**2.8**
BlurCar4	177.5	23.3	40.1	200.7	76.7	**21.6**	47.8
BlurCar2	169.2	31.0	224.3	260.2	67.1	161.9	**28.4**
FaceOcc2	61.4	19.5	13.4	15.9	72.4	**8.3**	14.7
Subway	140.2	146.4	101.8	9.4	11.7	141.2	**5.3**
Trans	108.1	74.0	96.9	**23.9**	122.3	35.4	43.3
Average	111.5	49.7	86.0	94.9	75.5	67.9	**23.7**

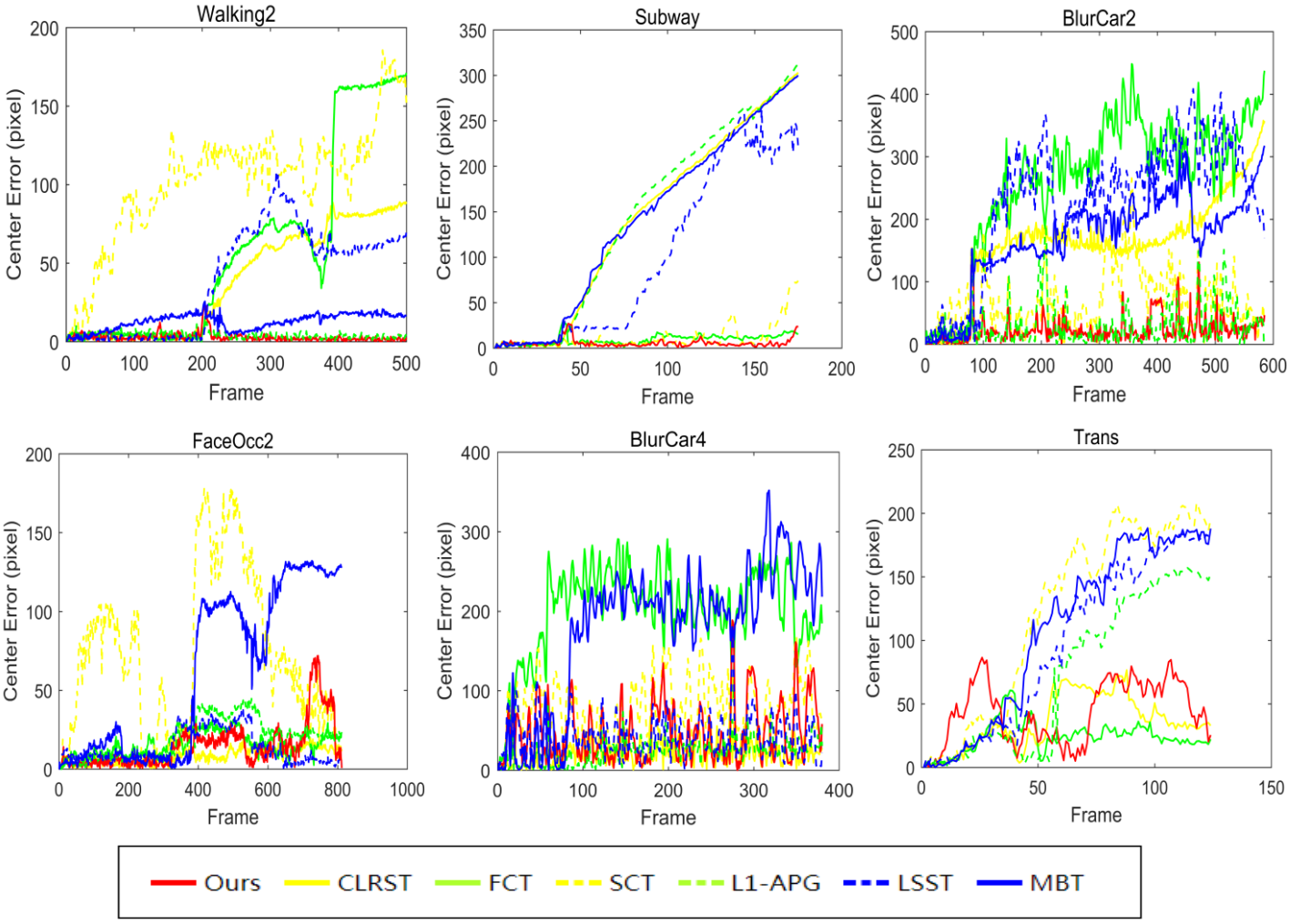

Fig2. Center location error (unit: pixel)

IV. Conclusion

In this paper, we propose the tracking algorithm combined the generative and discriminative method which was aiming at coping with the interference factors such as motion blur, occlusion and scale variation under complex scenes. At the same time, the weight of the target is taking into consideration when we select the best target. And the sum of production of the weight and two confidences were used to select the best target. Even if the target was polluted, it can also be robust in visual tracking. Combined with the pollution degree of the target area, when the pollution level is higher

than the given threshold, the target template set and classifier cannot be updated by the tracking target of the frame, so as to prevent the error accumulation which cause the losing of the target. By comparing the results of visual tracking in different scenes with different algorithms, the average overlap rate and the average center location error indicate that our algorithm has good effect and stability which can better deal with the unfavorable factors in video sequence, and it has higher accuracy and robustness in visual tracking.

Acknowledgment

The research is supported in part by the NSFC and Shanxi Provincial People's Government Jointly Funded Project of China for Coal Base and Low Carbon (No.U1510115, 51104157), the Qing Lan Project, the Jiangsu Province Natural Science Foundation of China (No. BK20150201). We gratefully acknowledge Academy of Finland, the Jorma Ollila Grant of Nokia Foundation, Central Fund of Finnish Cultural Foundation, the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research, the Ph.D. Programs Foundation of Ministry of Education of China (No. 20110095120008) and the China Postdoctoral Science Foundation (Nos.2013T60574, 20100481181).

References

1. Z. Chi, H. Li, H. Lu, and M.H. Yang, "Dual Deep Network for Visual Tracking," *IEEE Trans. Image Process.*, vol. 26, pp. 2005-2015, April 2017.
2. K. Zhang, Q. Liu, Y. Wu, and M.H. Yang, "Robust Visual Tracking via Convolutional Networks Without Training," *IEEE Trans. Image Process.*, vol. 25, pp. 1779-1792, April 2016.
3. F. Yang, H. Lu, and M.H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, pp. 1639-1651, April 2014.
4. C. Sun, F. Li, H. Lu, and G. H, "Visual Tracking via Joint Discriminative Appearance Learning," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 27, pp. 2576-2577, December 2017.
5. L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual Tracking with Fully Convolutional Networks," in IEEE International Conference on Computer Vision, 2016.
6 D. Wang, H. Lu, and C. Bo, "Visual Tracking via Weighted Local Cosine Similarity," *IEEE Trans. Cybernetics.*, vol. 45, pp. 1838-1850, September 2015.
7. T. Zhang, C. Xu, and M.H. Yang, "Multi-task Correlation Particle Filter for Robust Object Tracking," in IEEE Conference on Computer Vision and Pattern Recognition, 2017.
8. J. Gao, T. Zhang, X. Yang, and C. Xu, "Deep Relative Tracking," *IEEE Trans. Image Process.*, vol. 26, pp. 1845-1858, April 2017.
9. L. Zhang, H. Lu, D. Du, and L. Liu, "Sparse Hashing Tracking," *IEEE Trans. Image Process.*, vol. 25, pp. 840-849, Feberary 2016.
10. Y. Song, C. Ma, L. Gong, J. Zhang, R.W.H. Lau, and M.H. Yang, "CREST: Convolutional Residual Learning for Visual Tracking," in IEEE International Conference on Computer Vision, 2017.
11. C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in IEEE Conference on Computer Vision and Pattern Recognition, 2012.
12. T. Zhang, S. Liu, N. Ahuja, M.H. Yang, and B. Ghanem, "Robust Visual Tracking Via Consistent Low-Rank Sparse Learning," *Int. J. Comput. Vis.*, vol. 111, pp. 171-190, January 2015.
13. B. Ma, L. Huang, J. Shen, L. Shao, M.H. Yang, and F. Porikli, "Visual Tracking under Motion Blur," *IEEE Trans Image. Process.*, vol. 25, pp. 5867-5876, December 2016.
14. K. Zhang, L. Zhang, and M.H. Yang, "Real-time compressive tracking," in European Conference on Computer Vision, 2012.
15. X. Chen, W. Pan, J.T Kwok, and J.G Carbonell, "Accelerated Gradient Method for Multi-task Sparse Learning Problem," in IEEE International Conference on Data Mining, 2009.
16. H. Zhang, J. Yang, Y. Zhang, N.M. Nasrabadi, and T.S. Huang, "Close the loop: Joint blind image restoration and recognition with sparse representation prior," in International Conference on Computer Vision, 2011.
17. B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual Tracking Using Strong Classifier and Structural Local Sparse Descriptors," *IEEE Trans. Multimedia*, vol. 17, pp. 1818-1828, October 2015.
18 D. Wang, H. Lu, and M.H. Yang, "Least Soft-Threshold Squares Tracking," in IEEE Conference on Computer Vision and Pattern Recognition, 2013.
19. K. Zhang, L. Zhang, and M.H. Yang, "Fast Compressive Tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, pp. 2002-2015, October 2014.
20. Y. Wu, J. Lim, and M.H. Yang, "Online Object Tracking: A Benchmark," in IEEE Conference on Computer Vision and Pattern Recognition, 2013.