International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.5 (2014), pp.79-92
http://dx.doi.org/10.14257/ijsip.2014.7.5.08

Visual Tracking Based on Reversed Sparse Representation

Wenhui Dong

College of Physics and Electronic Engineering, Dezhou University, Dezhou 253023, China
dongwh_81@163.com

Abstract

In this paper, we propose a fast and robust tracking method based on reversed sparse representation. Be different from other sparse representation based visual tracking methods, the target template is sparsely represented by the candidate particles which are gotten by particle filter. In order to improve the robustness of the method, we use a target template set. Meanwhile, a two level competition mechanism is also introduced. In the first level, each target template is sparsely represented and all the candidate particles compete with each other by a similarity calculation, which is based on sparse coefficients. Then, the winners construct a target candidate set. In the second level, all the target candidates in the target candidate set compete with each other and the one which is the most similar to the template set is considered as the target. In addition, a template set update strategy is proposed to adapt the appearance variations of the target. Experimental results on challenging benchmark video sequences demonstrate that the proposed tracking algorithm performs favorably against several state-of-the-art methods.

Keywords: *Visual tracking, Sparse representation, Particle filter, Competition mechanism*

1. Introduction

Visual tracking is one of the important tasks in computer vision, as it can be widely applied in many fields [1-4]. Though it has been an active research area for several decades, it is still a challenge to design a robust tracking system, due to the intrinsic and extrinsic appearance variations of the target [5]. The intrinsic appearance variations are pose variation and shape deformation of the target. The extrinsic variations include changes resulting from different illumination, camera motion, and partial occlusion.

Current tracking method can be divided into two main categories: Generative methods and Discriminative methods. Discriminative methods usually train a classifier to separate the object from background. Because the data are coming in sequence, the classifier in these methods needs to be trained and updated online. Grabner, *et al.*, [6, 7] propose an online adaboost tracker, which selects the best discriminative features from the feature candidates pool. Saffari, *et al.*, [8] also proposes an online classifier for tracking, which is based on the online version of random forest algorithm. Generative methods use the explicit model to describe the target. These methods rely on the knowledge of the foreground and locate the object by finding a region with highest similarity. In reference [9], Zhou, *et al.*, construct an appearance adaptive model and cast the visual tracking as a particle filter framework. In [10], the changes of the target appearance are incrementally learned by an eigenspace representation and an online subspace learning algorithm is also proposed.

Recently, a new kind of generative method that achieves target tracking based on sparse representation has been extensively studied [11-13]. A typical example is the tracking method

ISSN: 2005-4254 IJSIP

proposed by Mei and Ling [12]. In their method, a sparse representation based appearance model for visual tracking is presented. The target appearance is expressed as a sparse linear combination with a basis library consisting of target templates and trivial templates via “11-minimization”. After that, tracking is continued using the particle filter framework. Finally, the tracking result is assigned to the observed sample that has the smallest reconstruction residual with the target templates and corresponding target coefficients. Although this method is effective, sparse representation needs to be calculated for each particle. Therefore the time cost is very large, which limits its performance. Although some works [14] have been proposed to speed up this tracker [12], they do not make much improvement.

In this paper, we propose a fast and robust tracking method based on reversed sparse representation. Be different from other sparse representation based visual tracking methods, the target template are sparsely represented by the candidate particles which is gotten by particle filter. Thus, the sparse representation only needs to be calculated once and the time cost is dramatically decreased. In order to improve the robustness of the method, we use a target template set and a two level competition mechanism. In the first level, each target template is sparsely represented and all the candidate particles compete with each other by a similarity calculation, which is based on sparse coefficients. Then, the winners construct a target candidate set, which is the output of the fist level competition. In the second level, all the target candidates in the target candidate set compete with each other and the one which is the most similar to the template set is considered as the target. In addition, a template set update strategy is proposed to adapt the appearance variations of the target.

The reminder of the paper is organized as follows. Section 2 gives the details of the reversed sparse representation. The proposed visual tracking with reversed sparse representation is described in Section 3. The experimental results are shown in Section 4, which demonstrate the effectiveness of the proposed tracking algorithm using some challenging videos. Finally, we conclude this work in Section 5.

2. Reversed Sparse Representation

Recently, sparse representation based methods are popular in visual tracking. The core idea of these methods is following the simple observation: A valid candidate can be sufficiently represented using only the target templates of the same class. So they use the target templates and trivial templates to construct the dictionary and each candidate is sparsely represented on it. If the candidate is valid, the sparse coefficients of the target templates will be much larger than those of trivial templates. Because it is a linear representation problem, we can also consider it reversely that a target template can also be sparsely represented on the candidates set, if there is at least one valid candidate in it. The sparse coefficients of the valid candidates will be larger than those of others. In order to be different from other methods, we call it reversed sparse representation.

Given a template $t \in R^d$ and N corresponding candidates $\{x_i, i = 1, 2, \cdots, N\}$ (at least one valid candidate in it), the template t can be represented as

$$t = X\alpha \quad (1)$$

Where, $X = [x_1, x_2, \cdots x_N]$ is a $d \times N$ ($d << N$) candidate matrix, which contains N candidates. $\alpha = [\alpha_1, \alpha_2, \cdots \alpha_N]^T$ is the sparse coefficient vector. As can be seen, the linear equation in (1) is underdetermined. Then a l_1 regularization term is added to solve this problem [12].

$$\min \left\| X\alpha - t \right\|_2^2 + \lambda \left\| \alpha \right\|_1 \tag{2}$$

Where, $\| \|_2$ and $\| \|_1$ are the l_2 norm and l_1 norm respectively.

The reserved sparse representation is especially useful for visual tracking. Firstly, the sparse coefficient vector α reflects the relation between the template and the candidates. The largest one usually corresponds to the target. So the similarity can be calculated based on it. Figure 1 gives an illustration, which demonstrates a good similarity measurement and an excellent distinguishing capability. Secondly, it only needs to calculate the sparse representation once and the time cost is dramatically decreased. If the method in reference [12] is used, N times sparse decompositions are required. Suppose each decomposition needs m iterations and each iteration costs t_0 seconds, then the sparse presentation in [12] will cost $N \times m \times t_0$ seconds. If under same condition, the sparse presentation of our method will cost $m \times t_0$ seconds for one template. Suppose there are T templates, then the total cost will $m \times t_0 \times T$. Because $N \gg T$, our method is obviously timesaving and efficient.

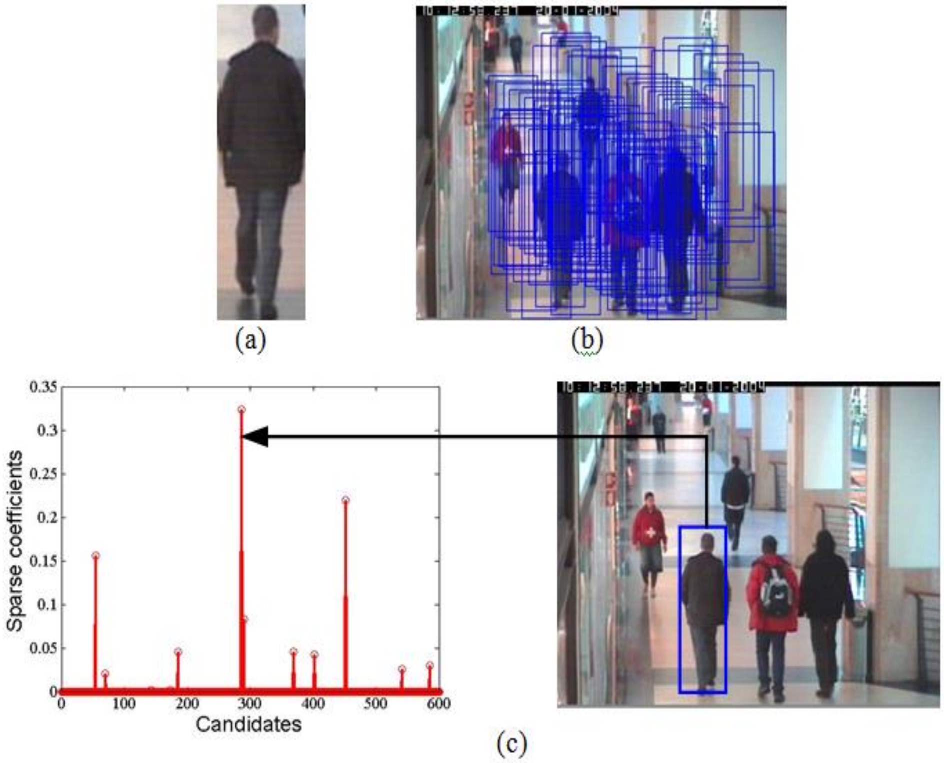

Figure 1. The Reversed Sparse Representation, (a) Target Template, (b) Candidates in the Frame, (c) The Largest Coefficient is Corresponding to the Target

3. Visual Tracking Based on Reversed Sparse Representation

Given a target template and candidates which are gotten by particle filter, the reversed sparse representation can be naturally used for tracking. However, in order to improve the robustness of the tracking, we use a target template set and a two level competition mechanism. In the first level, each target template is sparsely represented and all the candidate particles compete with each other by a similarity calculation, which is based on sparse coefficients. Then, the winners construct a target candidate set. In the second level, all the target candidates in the target candidate set compete with each other and the one which is the most similar to the template set is considered as the target. In addition, a template set update strategy is proposed to adapt the appearance variations of the target. The whole framework is demonstrated in Figure 2.

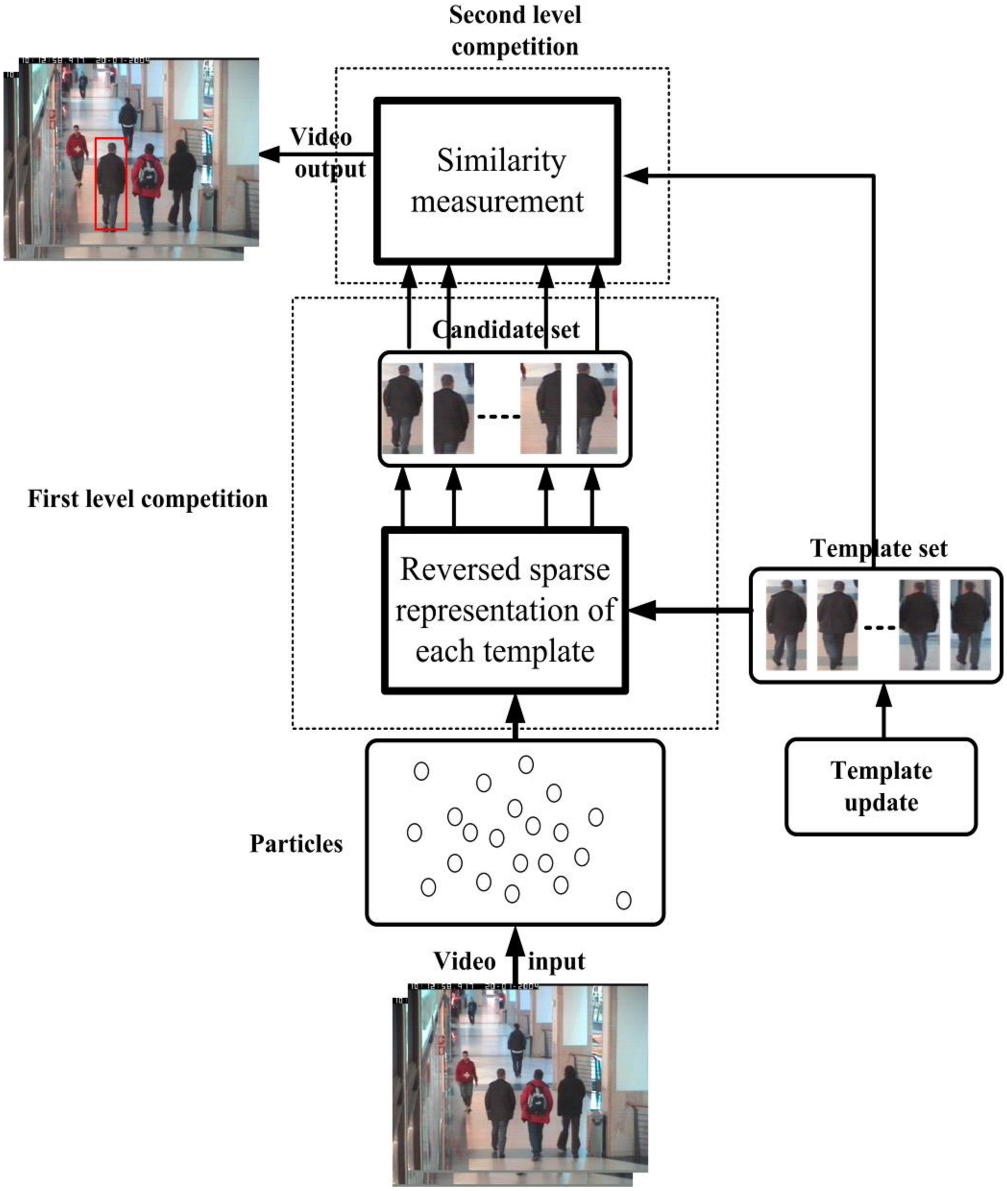

Figure 2. The Framework of the Proposed Visual Tracking Method

3.1. First Level Competition

In the first level competition, each target template is sparsely represented based on reversed sparse presentation and all the candidate particles compete with each other by a similarity calculation, which is based on reversed sparse coefficients. Then, the winners construct a target candidate set, which is the output of the first level competition.

Particle filter is used in the first level competition, which needs two components: motion model $p(X_t \mid X_{t-1})$ and observation model $p(Z_t \mid X_t)$. The motion model describes the state transition between two consecutive frames and products many particles to decide where and how we extract the candidates. In this paper, we use an affine image warping which can transform the coordinate and center the image to model the target motion. Six parameters x_t, y_t, θ_t, s_t, α_t, φ_t (x translation, y translation, rotation angle, scale, aspect ratio, and skew direction at time t) are used to represent the state variable $X_t = [x_t, y_t, \theta_t, s_t, \alpha_t, \varphi_t]$ .All parameters are independent and can be modeled in Gaussian distribution around X_{t-1}. Namely,

$$P(X_t \mid X_{t-1}) = N(X_t, X_{t-1}, \Sigma) \tag{3}$$

Where, Σ is a diagonal covariance matrix which contains $\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2$, and σ_φ^2.

Observation model $P(Z \mid X)$ reflects the similarity between the target candidate and the template. In this paper, the reversed sparse coefficients are used to construct the observation model. Given a template and N candidates extracted by the motion model, the reversed sparse coefficient of each candidate can be gotten. We denote the coefficient as α_i and define the similarity measurement as:

$$S_i = \frac{\alpha_i}{\sum_{j=1}^{N} \alpha_j} \tag{4}$$

The observation model can be gotten as:

$$P(Z \mid X) = e^{S_i} \tag{5}$$

For a template, N candidates extracted by the motion model compete with each other using (5).The largest one is the winner and is considered as a best candidate corresponding to the template. Suppose there are T templates in the template set, N candidates will compete T times. Finally, a target candidate set can be obtained with T candidates. Figure 3 demonstrates the first level competition.

3.2. Second Level Competition

After the first level competition, all the T target candidates in the target candidate set compete with each other again and the one which is the most similar to the template set is considered as the target.

Define Similarity with the object nearest neighbor:

$$S(c, T_e) = \max_{t_i \in T_e} S(c, t_i) \tag{6}$$

Where, c represents the target candidate, T_e is the template set. The formula measures the similarity between the target candidate and the template set. Any similarity function can be used, such as $\cos(\theta)$ (θ is the angle between the two normalized vectors).

Then, the final target is chosen by

$$\hat{Y} = \arg\max_{C} \{ S(c_i, T_e), | i = 1, 2, \cdots T \} \tag{7}$$

Where, C is the target candidate set which contains T target candidates.

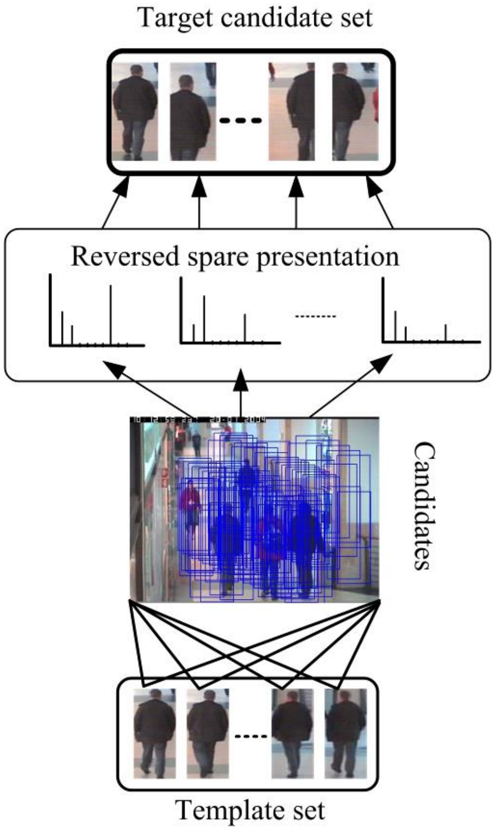

Figure 3. The First Level Competition

3.3. Template Update

Because of the inevitable appearance variations caused by factors such as illumination and pose change, the template update is needed. Motivated by reference [13], we dynamically update the template set in this paper. Firstly, a probability sequence is generated according to the number of templates T .

$$L_p = \{0, \frac{1}{2^{T-1}-1}, \frac{3}{2^{T-1}-1}, \frac{7}{2^{T-1}-1}, \cdots, 1\} \tag{8}$$

We order the templates by their obtaining time and assign a corresponding element in L_p to each one according to the order. Secondly, we generate a random number on the interval [0, 1] in uniform distribution. Finally, the template to be replaced is chosen according to the section that the random number lies in. This update mechanism can keep the earlier templates longer than the new templates, which reduces the drifting.

However, the template set will also degrade if we use some imprecise samples to update it. Given the tracking result $\hat{Y}$ and the template set T_e, we evaluate $\hat{Y}$ by the similarity with the object nearest neighbor:

$$S_{\max}(\hat{Y}, T_e) = \max_{t_i \in T_e} S(\hat{Y}, t_i) \tag{9}$$

$$S_{\min}(\hat{Y}, T_e) = \min_{t_i \in T_e} S(\hat{Y}, t_i) \tag{10}$$

Two thresholds τ_1 and τ_2 $(\tau_1 > \tau_2)$ are used. If $S_{\min}(\hat{Y}, T_e) > \tau_1$, the tracking result is so similar to the template set that template update is not needed. If $S_{\max}(\hat{Y}, T_e) < \tau_2$, the tracking result is too different to be used to update the template set. If $\tau_2 < S_{\max}(\hat{Y}, T_e) < \tau_1$ & $\tau_2 < S_{\min}(\hat{Y}, T_e) < \tau_1$, the tracking result $\hat{Y}$ is used to update the template set. Algorithm 1 summarizes the template update.

Algorithm 1: Template Update

```
Input: The new tracking result $\hat{Y}$ ,Template set $T_e = \{t_i \mid i = 1, 2, \cdots, T\}$
1 Generate the probability sequence in (8).
2 Generate a random number on the interval [0, 1] in uniform distribution.
3 Evaluate $\hat{Y}$ using (9) and (10).
4 If $\tau_2 < S_{\max}(\hat{Y}, T_e) < \tau_1$ & $\tau_2 < S_{\min}(\hat{Y}, T_e) < \tau_1$, then replace the chosen template by $\hat{Y}$ ;
Else, don't update the template set.
Output: New template set
```

3.4. Summary of the Proposed Visual Tracking Algorithm

When a new frame comes, candidates are extracted using the motion model in (3).Each template is sparsely represented on the candidates and the similarity can be calculated by (5). Then, T target candidates that recommended by the T templates are evaluated by (6). Finally, the tracking target is chosen by (7). In addition, template set is updated in order to adapt to the appearance variations of the target. The whole tracking algorithm is summarized in Algorithm 2.

Algorithm 2: Visual Tracking based on Reversed Sparse Representation

```
Input: The initial state of the target $X_0 = [x_0, y_0, \theta_0, s_0, \alpha_0, \varphi_0]$ and $T$ templates
1 Sample the candidates according to the motion model in (3).
2 Sparsely represent each template on the candidates and obtain the likelihood under the
  observation model (5).
3 Evaluate $T$ target candidates by (6).
4 Estimate the current state of the target $X_t$ according to (7).
5 Update the template set using Algorithm 1.
6 Go to step 1 until the last frame.
Output: The current state of the target $X_t$
```

4. Experimental Results

We evaluate the performance of our method from two aspects: time cost and the accuracy. Five challenging videos are used and the details are demonstrated in Table 1. Figure 4 also shows the tracking target in the five videos.L1 tracker [12], IVT tracker [15] and MIL tracker [16] are also used here for comparation. Firstly, we compare the proposed method with the L1 tracker in the aspect of time cost. Then, all three state-of-the-art trackers are used to compare the accuracy.

Table 1. Challenging Videos

Video	Frame	Main challenge
Car4[12]	659	Illumination variation, scale change
David[15]	462	Moving camera, illumination variation
Faceocc2[13]	819	Occlusions, heavy appearance variation
Sylvester[16]	900	3D motion, illumination variation
ThreePastShop2cor[17]	351	Occlusions, scale change

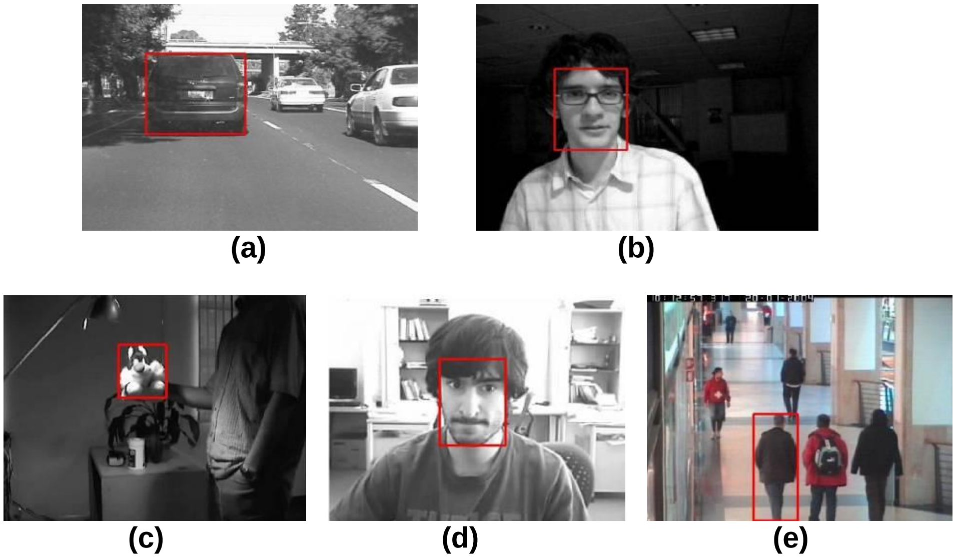

Figure 4. Targets Tracked in the Videos, (a) Car 4 (b) David (c) Sylvester (d) Faceocc2, (e) Three Past Shop 2 cor

4.1. Evaluation the Time Cost of the Trackers

In this experiment, we evaluate the time cost of the L1 tracker and the proposed tracker. The two trackers are both based on spare representation. The main difference is that L1 tracker uses target templates and trivial templates as the dictionary and each candidate is sparsely represented on it, while our tracker uses the candidates as the dictionary and each target template is sparsely represented on it. Car4 sequence is used here. In both trackers, we use 1024 particles and 20 templates. The measurement is implemented on the computer with 3G CPU and 3G RAM. Figure 5 shows the tracking time cost of the two methods. As can be seen, the proposed method is much faster than the L1 tracker. The average time cost of our tracker is 4.2seconds for one frame, while the average time cost of L1 tracker is about 30 seconds.

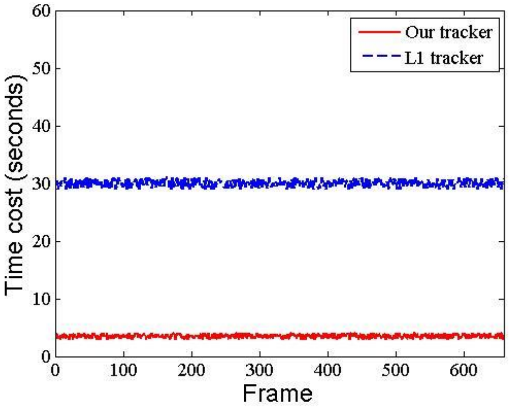

Figure 5. Time Cost of the Two Trackers

4.2. Evaluation the Accuracy of the Trackers

In this experiment, four challenging videos are used to evaluate the accuracy of the four trackers (L1 tracker, IVT tracker, MIL tracker and our tracker). For fair evaluation, we use the source codes of the three trackers provided by the authors. Each tracker runs with their parameters. Four videos (David, Sylvester, Faceocc2, and ThreePastShop2cor) are used here. The tracking errors in each frame are calculated as the distance between the ground truth and the center of the tracking rectangle. Some result images are shown, in which the boxes in red, black, blue and green are corresponding to the proposed tracker, L1 tracker, MIL tracker and IVT tracker respectively.

David sequence is used to evaluate the performance of the trackers when the target undergoes illumination variations, in which the boy walks out of the dark room and into an area with spot lights. Figure 6 shows some exemplar frames of the tracking results and Figure 7 demonstrates the tracking error in each frame. As can be seen, the MIL tracker and L1 tracker drift far away from frame 200. Our tracker and the IVT tracker do a good work. Although the tacking errors of our tracker is larger than the IVT tracker in some occasions after frame 300, our tracker still has the lowest mean error(Table 2).

Table 2. Mean Tracking Errors of the Videos

	David	Faceocc2	Sylvester	ThreePastShop2cor
MIL tracker	45.66	30.88	45.69	88.21
IVT tracker	11.39	27.88	49.73	43.82
L1 tracker	68.99	19.33	39.45	38.17
Proposed tracker	9.54	12.02	20.25	25.25

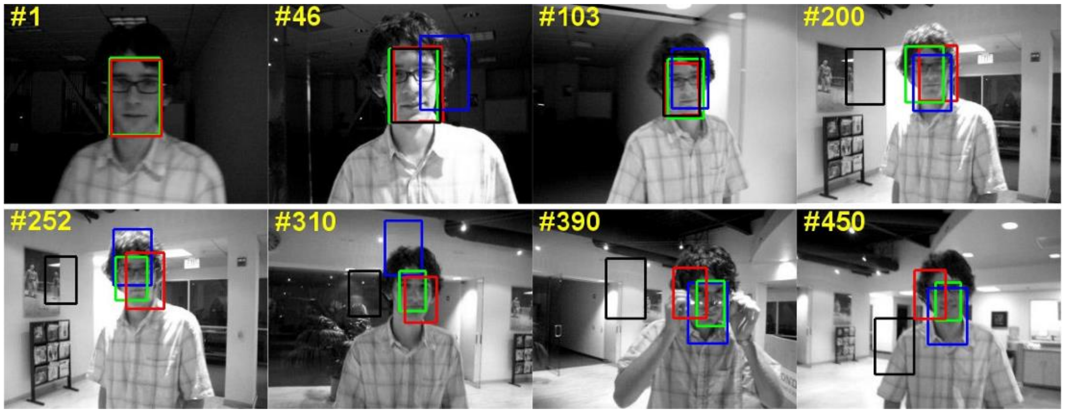

Figure 6. Tracking Results of David Sequence

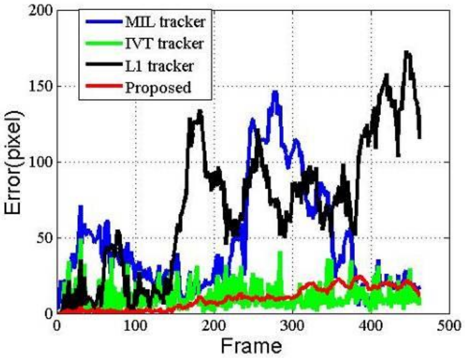

Figure 7. Tracking Errors of David Sequence

The Sylvester sequence in which the target has pose variations is also used to evaluate the trackers. The tracking results and tracking errors are shown in Figure 8 and Figure 9 respectively. From the results, we can see that the proposed tracker tracks the doll successfully when large variation happens, while other three trackers drift away. Failures are observed from frame 391 for IVT tracker, frame 101 for L1 tracker and frame 300 for MIL tracker respectively. The mean tracking error of our tracker is also the lowest among the four trackers. The results of this video demonstrate that our tracker can deal with the large pose variations of the target.

We use the Faceocc2 and ThreePastShop2cor sequences to evaluate the tracker when there is occlusion. In Faceocc2 sequence, The L1 tracker tracks the target in the first 300 frames, but then drifts away and fails to locate the face at frame 451.The IVT tracker and MIL tracker also drift away when long duration occlusion occurs. Our tracker still does best in this video. Its mean tracking error is 12.02, which is the lowest among the four trackers. The ThreePastShop2cor is a very challenge sequence, in which the target is occluded completely from frame 107 to frame 120. After the heavy occlusion, other three trackers drift far away and begin to track wrong target. Although our tracker also drifts away, it still tracks partial of the target. The tracking errors in Figure 13 also demonstrate that the proposed tracker has the lowest tracking error.

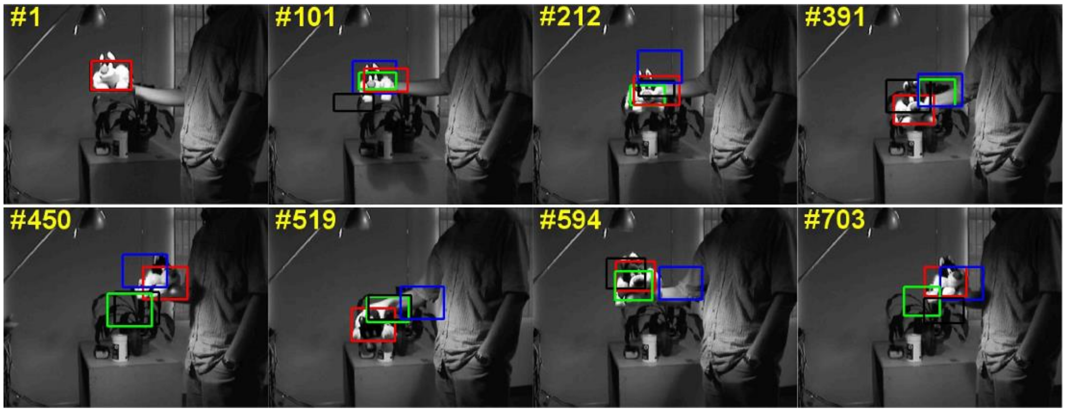

Figure 8. Tracking Results of Sylvester Sequence

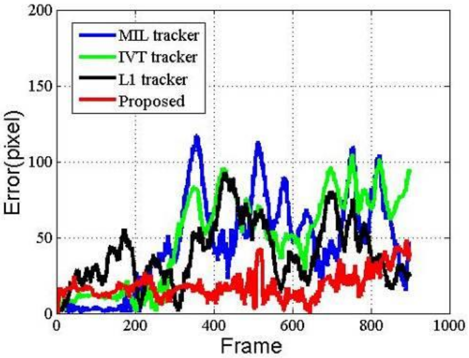

Figure 9. Tracking Errors of Sylvester Sequence

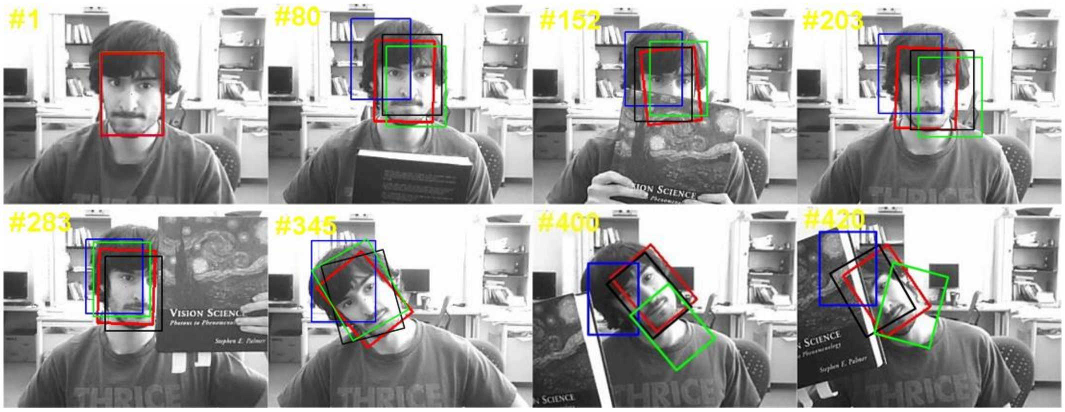

Figure 10. Tracking Results of Faceocc2 Sequence

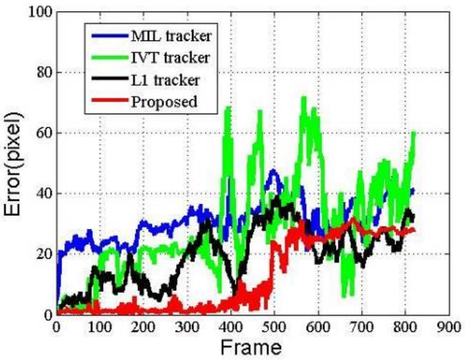

Figure 11. Tracking Errors of Faceocc2 Sequence

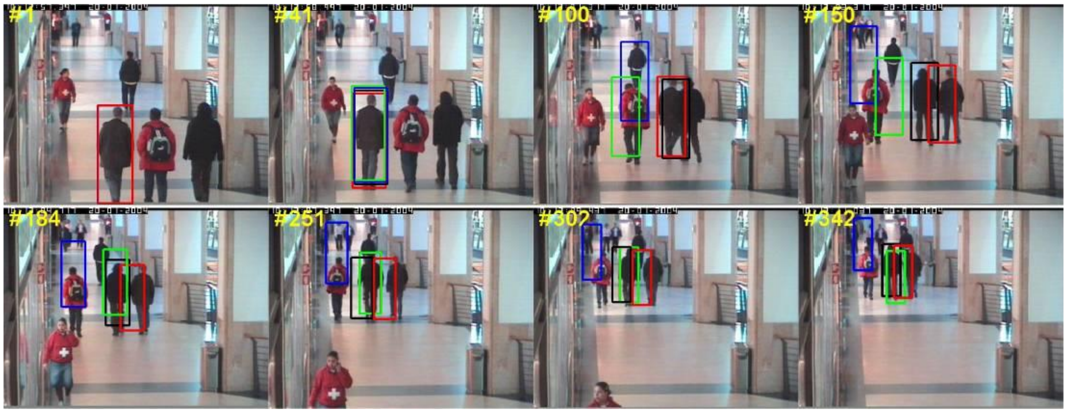

Figure 12. Tracking Results of ThreePastShop2cor Sequence

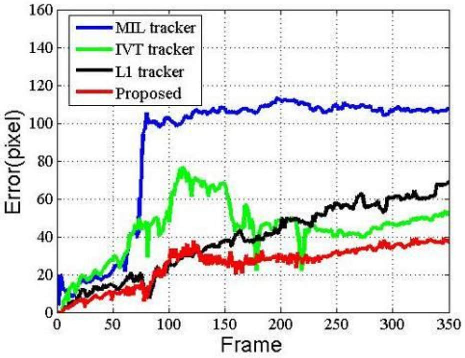

Figure 13. Tracking Errors of ThreePastShop2cor Sequence

5. Conclusion

In this paper, we propose a fast and robust tracker based on reversed sparse representation. Be different from the state-of-the-art trackers based on sparse representation, the target template is sparsely represented by the candidate particles which is gotten by particle filter. In order to improve the robustness of the method, a target template set and a two level competition mechanism are introduced. In addition, a template set update strategy is proposed to adapt the appearance variations of the target. The tracking results on challenging sequences demonstrate that the proposed tracking algorithm performs favorably against several state-of-the-art methods. Although our proposed tracker does well in the experiments, drifts are observed in some extreme conditions. This should be investigated in the future work.

References

[1] S. Kim, H. S. Choi, K. M. Yi, J. Y. Choi and S. G. Kong, "Intelligent Visual Surveillance-A Survey", International Journal of Control, Automation, and Systems, vol. 8, no. 5, **(2010)**, pp. 926-939.

[2] A. Yilmaz, O. Javed and M. Shah, "Object tracking: A survey", ACM Computing Surveys, vol. 38, no. 4, **(2006)**, pp.1-45.

[3] A. Bakhtari, M. Mackay and B. Benhabib, "Active-vision for the autonomous surveillance of dynamic, multi-object environments", Journal of Intelligent Robot System, vol. 54, no. 4, **(2009)**, pp. 567–593.

[4] V. Jelača, A. Pižurica, J. O. Niño-Castañeda, A. Frías-Velázquez and W. Philips, "Vehicle matching in smart camera networks using image projection profiles at multiple instances", Image and Vision Computing, vol. 38, **(2013)**, pp. 673-685.

[5] H. X. Yang, L. Shao, F. Zheng, L. Wang and Z. Song, "Recent advances and trends in visual tracking: a review", Neurocomputing, vol. 74, no.18, **(2012)**, pp. 3823-3831.

[6] H. Grabner and H. Bischof, "On-line boosting and vision", Computer Vision and Pattern Recognition, **(2006)**, pp. 260-267.

[7] H. Grabner and H. Bischof, "Real-time tracking via on-line boosting", Proceedings 17th British Machine Vision Conference, **(2006)**, pp. 47-56.

[8] A. Saffari, C. Leistner, J. Santner, M. Godec and H. Bischof, "On-line random forests", IEEE 12th International Conference on Computer Vision Vorkshops, **(2009)**, pp. 1393-1400.

[9] S. Zhou, R. Chellappa and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters", IEEE Transactions on Image Processing, vol. 13, no.11, **(2004)**, pp. 1491–1506.

[10] X. Li, W. Hu, Z. Zhang, X. Zhang and G. Luo, "Visual tracking via incremental log-Euclidean Riemannian subspace learning", Computer Vision and Pattern Recognition, **(2008)**, pp. 1-8.

[11] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma. "Robust Face Recognition via Sparse Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 1, **(2009)**, pp. 210-227.

[12] X. Mei and H. Ling, "Robust Visual Tracking and Vehicle Classification via Sparse Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 11, **(2011)**, pp. 2259-2272.

[13] X. Jia, H. Lu, M.-H. Yang, "Visual Tracking via Adaptive Structural Local Sparse Appearance Model", IEEE Conference on Computer Vision and Pattern Recognition, **(2012)**, pp. 1822-1829.

[14] X. Mei, H. Ling, Y. Wu, E. Blasch and L. Bai, "Minimum error bounded efficient L1 tracker with occlusion detection," IEEE Conference on Computet Vision and Pattern Recognition, **(2011)**, pp. 1257-1264.

[15] A. Ross, J. Lim, R.-S. Lin and M.-H. Yang, "Incremental learning for robust visual tracking", International Journal of Computer Vision, vol. 77, no. 3, **(2008)**, pp. 125–141.

[16] B. Babenko, M.-H. Yang and S. Belongie, "Visual Tracking with Online Multiple Instance Learning", IEEE Conference on Computer Vision and Pattern Recognition, **(2009)**, pp. 983-990.

[17] "CAVIAR [Online]", Available: http://groups.inf.ed.ac.uk/vision/ CAVIAR/CAVIARDATA1/.

Authors

Wenhui Dong, she received her B.S. degree in electronic engineering from Qufu Normal University in 2003 and M.S. degree in communication and information systems from Shandong University in 2006.Now she is a lecturer of College of Physics and Electronic engineering, Dezhou University. She focuses on computer vision, pattern recognition and image processing.