

Visual Tracking with Fragments-Based PCA Sparse Representation

Peishu Qu

*College of Physics and Electronic engineering, Dezhou University,
Dezhou 253023, China
qupsh@163.com*

Abstract

In this paper, we propose a robust tracking method with a novel appearance model based on fragments-based PCA sparse representation. It samples non-overlapped local image patches within the templates in PCA subspace. Then, the candidate local image patches are sparse represented by the local template patches in PCA subspace. Finally, tracking is continued using the particle filter for propagating sample distributions over time. In addition, the templates are updated online based on incremental subspace learning. Using the fragments-based PCA templates rather than the image templates facilitates the tracker to handle significant illumination and pose change as well as occlusion. Experimental results on challenging videos show that our method can track accurately and robustly, and outperform many other state-of-the-art trackers.

Keywords: *Visual tracking, Sparse representation, PCA subspace, Particle filter*

1. Introduction

Visual tracking is one of the essential and fundamental tasks in computer vision as it can be widely applied in video surveillance, vehicle navigation, human–computer interaction, *etc.*, [1-4]. However, designing a robust tracking system still remains a challenge for the community, due to the appearance variations of the target. The appearance variations include intrinsic variations and extrinsic variations [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. So a robust appearance model that can deal with the aforementioned difficulties is critical to the tracking algorithm and many appearance models have been proposed recently. In reference [6], an adaptive fragments-based appearance model fusing color histogram and SIFT features is proposed, which can solve the problems of color similar target, scale change and occlusion in target tracking. Ross *et al.*, [7] represent the target object by a low dimensional PCA subspace and propose an online update approach for efficiently learning and updating the appearance model. Experimental results illustrate that this method is effective in handling in-plane rotation, scale, illumination variation and pose change. However, it has also been shown that the scheme is sensitive to partial occlusion. Recently, sparse representation has been extensively studied and applied in visual tracking. This kind of methods estimates the target appearance by finding a sparse linear combination over a basis library containing target and trivial templates. In reference [8], the sparse representation of the candidate target is solved by l_1 minimization. Although this method appears to promise robustness in visual tracking, it only uses training templates directly cropped from the images, which makes it difficult to handle significant view and pose changes. More recently, Xu Jia *et al.*, [9] propose a method based on the structural local sparse appearance model, which exploits both partial information

and spatial information of the target. This method is successful in the occluded cases but it is not clear if such a method is effective when the target undergoes significant pose variations.

In this paper, we propose a robust tracking method with a novel appearance model. The appearance model is based on fragments-based PCA sparse representation. It samples non-overlapped local image patches within the templates in PCA subspace as the sparse codes. Then, the candidate local image patches are sparse represented. Finally, tracking is continued using the particle filter for propagating sample distributions over time. In addition, the templates are updated online based on incremental subspace learning. Using the fragments-based PCA templates rather than the image templates facilitates the tracker to handle significant illumination and pose change as well as occlusion.

The remainder of the paper is organized as follows. Section 2 reviews the related works. The proposed visual tracking with fragments-based PCA 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. Related Works

Much work has been done in appearance modeling and representation that aims at designing a robust visual tracking system. In this section, we discuss the most relevant algorithms: subspace representation and sparse representation methods.

2.1. Subspace Representation

Subspace representation aims at adapting the appearance variations with a low-dimensional subspace. These methods are effective to model the object appearance undergoing pose and illumination changes for visual tracking. Ross *et al.*, [7] propose an incremental visual tracking (IVT) method which introduces an online update approach for efficiently learning and updating a low dimensional PCA subspace representation of the target. Other sophisticated subspace representation methods (graph-based learning subspace [10, 11], and data-driven constrained subspace [12]) are proposed to find a more optimal subspace for performance improvement. Although these methods are effective in handling pose and illumination changes for visual tracking, they can not handle occlusions and will suffer from failure caused by occlusion in a long duration. In PCA subspace, the target is represented by equation (1).

$$y = Uz + e \quad (1)$$

Where, y denotes the target, z is the corresponding coefficient vector, U represents the matrix of column basis vectors, and e is the error.

In PCA, the error vector e is considered as Gaussian distributed with small variances. Therefore, the coefficient vector z and the reconstruction error e_{rec} can be approximated by equation (2) and (3). When the target is partially occluded, the noise term cannot be modeled with small variance. Hence, the PCA subspace representation is sensitive to partial occlusion.

$$z = U^T y \quad (2)$$

$$e_{rec} = \|y - UU^T y\|_2^2 \quad (3)$$

2.2. Sparse Representation

Sparse representation indicates another path to model the appearance of the object. Initially inspired by the work on robust face recognition in reference [13], Mei *et al.*, [8] propose a sparse representation based appearance model for visual tracking, in which the target appearance is expressed as a sparse linear combination with a basis library consisting of target templates and trivial templates. Given a target template set $T = [t_1, t_2, \dots, t_n] \in R^{d \times n}$, ($d \geq n$), the candidate object y can be represented by

$$y = [T, I] \begin{bmatrix} \omega \\ e \end{bmatrix} = A c \quad (4)$$

Where $I = [i_1, i_2, \dots, i_d]$ is the $d \times d$ identity matrix, which contains d trivial templates. Each trivial template has only one nonzero element. $\omega = [\omega_1, \omega_2, \dots, \omega_n]^T$ is the target coefficient, $e = [e_1, e_2, \dots, e_d]^T$ is the trivial coefficient vector. Then a l_1 regularization term is added to solve equation (4).

$$\min \|A c - y\|_2^2 + \lambda \|c\|_1 \quad (5)$$

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

The tracking result is then assigned to the observed sample with the smallest reconstruction residual.

$$\varepsilon(y) = \|y - T \omega\|_2^2 \quad (6)$$

The method only considers the holistic representation and do not make full use of the sparse coefficients. In reference [9], Xu Jia *et al.*, propose a structural local sparse appearance model, and exploits both partial information and spatial information of the target. The method helps locate the target more accurately and handle partial occlusion.

2.3. Motivation of our Method

Our method exploits the strength of both subspace learning and sparse representation for modeling object appearance. This method samples non-overlapped local image patches within the templates in PCA subspace. The candidate local image patches are sparse represented by the local template patches in PCA subspace. In addition, the templates are updated online based on incremental subspace learning. Our method has the following advantage. First, compared with other sparse represent method that using image templates, our method exploits the advantage of subspace representation and can handle the undergoing pose and illumination changes for visual tracking. Second, by introducing fragments-based method to sparse representation, our method exploits the spatial information as well as the partial information, which helps it more effective in handling occlusion.

3. Visual Tracking with Fragments-Based PCA Sparse Representation

3.1. Fragments-Based PCA Sparse Representation for Target Appearance Model

Given a set of target images $I = \{I_1, I_2, \dots, I_n\}$ in initialization, we can get the templates in PCA subspace by computing the eigenvectors U of the covariance matrix I_{cov} .

$$\begin{cases} I_{\text{cov}} = \frac{1}{n-1} \sum_{i=1}^n (I_i - \bar{I})(I_i - \bar{I})^T \\ \bar{I} = \frac{1}{n} \sum_{i=1}^n I_i \end{cases} \quad (7)$$

\bar{I} is the mean of the images. The eigenvectors U can be calculated by the singular value decomposition. Figure 1 shows the target images and the templates in PCA subspace.

Each template in U is sampled to M local non-overlapped patches. These local patches are used as the dictionary to encode the candidate target. Given the template set $U = \{u_1, u_2, \dots, u_n\}$ and a candidate y , we sample each template and can get dictionary $P = [p_1, p_2, \dots, p_{n \times M}] \in R^{p_d \times (n \times M)}$. For the candidate y , we also sample it in the same way and get $y = [y_1, y_2, \dots, y_M] \in R^{p_d \times M}$, where p_d is the dimension of the local patch, n is the number of templates. With the sparsity assumption, the local patches of the candidate target can be represented as the linear combination of the dictionary set by solving equation (8).

$$\min_{c_i} \|y_i - P c_i\|_2^2 + \lambda \|c_i\|_1 \quad (8)$$

Where, c_i is the sparse code of the patch y_i . For the whole candidate y , we can get all the sparse coefficients $C = [c_1, c_2, \dots, c_M]$. Before solving optimization problem in formula (8), all the vectors need to be l_2 normalized. Otherwise the decomposition coefficients would be sensitive to the magnitudes of the vectors.

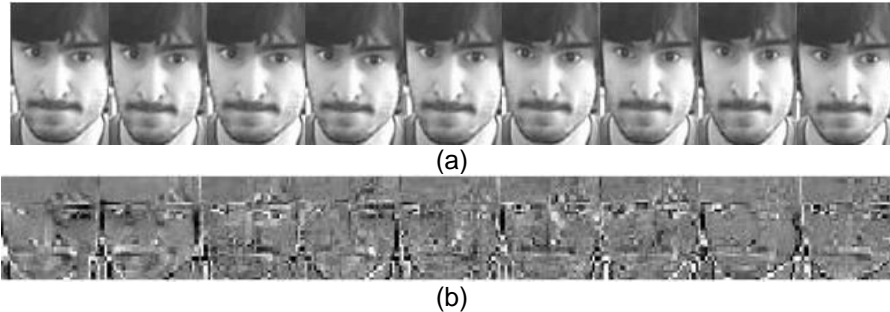


Figure 1. The Target Images and the Templates in PCA Subspace: (a) Target Images, (b) Templates in PCA Subspace

For a candidate patch y_i , it will be best described by the patches at the same position of the templates. That means the sparse coefficient $c_i = [c_{i1}, c_{i2}, \dots, c_{i(n \times M)}]^T$ will get the large elements in the corresponding locations of the candidate patch and very small elements in other locations. Figure 2 gives an example. We divide c_i into n segment according the template that the element corresponds to and covert the sparse coefficient c_i to the matrix c'_i . Each row in c'_i represents a template in U and each column represents a local patch of the template.

$$c'_i = \begin{bmatrix} c_{i1}, & c_{i2}, & \dots & c_{iM} \\ c_{i(M+1)}, & c_{i(M+2)}, & \dots & c_{i(2M)} \\ \dots & \dots & \dots & \dots \\ c_{i((n-1)M+1)}, & c_{i((n-1)M+2)}, & \dots & c_{i(nM)} \end{bmatrix} \quad (9)$$

Then, we calculate the mean value of each column in c'_i and obtain the vector V_i . For all the patches in the candidate y , a matrix $V = [V_1; V_2; \dots; V_n]$ can be obtained and the diagonal elements of V can be used to represent the candidate. By doing this the influence of outliers is reduced and the structural information is retained in the representation to better locate the target.

3.2. Particle Filter Framework for Visual Tracking

We embed the fragments-based PCA sparse representation appearance model into the particle filter framework to form a robust tracking algorithm. Let X_t is the state variable of the object at time t , $Z_{1:t-1} = \{Z_1, Z_2, \dots, Z_{t-1}\}$ denotes all the observations up to time $t-1$. Then, the state vector X_t can be calculated using the Bayes rule.

$$p(X_t | Z_{1:t}) = \frac{p(Z_t | X_t) \int p(X_t | X_{t-1}) p(X_{t-1} | Z_{1:t-1})}{p(Z_t | Z_{1:t-1})} \quad (10)$$

Where, $p(Z_t | X_t)$ denotes the observation likelihood, $p(X_{t-1} | Z_{1:t-1})$ can be approximated by the weighted sample set $\{(X_t^i, \omega_t^i) | i = 1, 2, \dots, n\}$, ω_t^i is the weight.

The weight can be updated by the following formula:

$$\omega_t^i = \omega_{t-1}^i \frac{p(Z_t | X_t^i) p(X_t^i | X_{t-1}^i)}{q(X_t | X_{1:t-1}, Z_{1:t})} \quad (11)$$

Commonly, the motion model $q(X_t | X_{1:t-1}, Z_{1:t}) = p(X_t | X_{t-1})$. In this paper, we model the target motion in the two consecutive frames as an affine image warping, which transforms

the coordinate and centers the image. The state variable X_t is modeled by the six parameters. $X_t = [x_t, y_t, s_t, \theta_t, \alpha_t, \varphi_t]$, where, $x_t, y_t, s_t, \theta_t, \alpha_t, \varphi_t$ denote x translation, y translation, scale, rotation angle, aspect ratio, and skew direction at time t . Each parameter is independent and modeled in a Gaussian distribution around the counterpart in X_{t-1} . Namely,

$$P(X_t | X_{t-1}) = N(X_t, X_{t-1}, \Sigma) \quad (12)$$

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

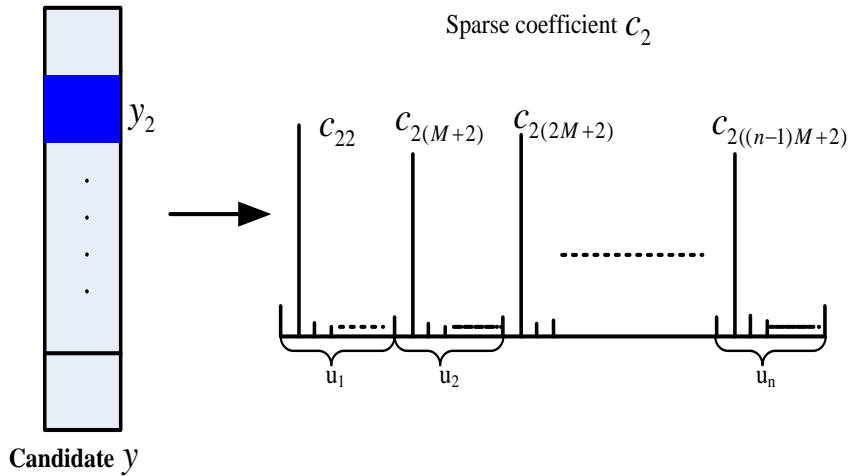


Figure 2. Sparse Coefficient of the Candidate Patch y_2

Then, the weight ω_t^i in equation (11) becomes the observation model $p(Z | X)$. In our method, we use the diagonal element of V to construct the observation model.

$$p(Z | X) = e^{-\sum_{k=1}^n f_k} \quad (13)$$

Where, f_k denotes the diagonal element of V .

3.3. Template Update

It is essential to update the eigen templates for handling appearance changes. However, if some imprecise samples are used, the model will degrade. In this paper, we use the ratio in equation (14) to detect the occlusion. If certain patch i in the target is occluded, the sparse coefficients of it will be dense and the value $\eta(i)$ in (14) will be small. If $\eta(i) < tr$, (tr is the threshold), then it is occluded. The model will not be updated as long as there is one patch is occluded.

$$\eta(i) = \frac{f_k(i)}{\sum_{j=1}^M V_i(j)} \quad (14)$$

Where, $f_k(i)$ denotes the diagonal element corresponding patch i , $V_i(j)$ is the j -th sparse coefficient in vector V_i .

If the whole target is not occluded, it will be used to update the model. In our approach, we employ incremental learning procedure in [7] to update the template set. We briefly introduce the update algorithm to make this paper self-contained.

Suppose we have an image set $A = [I_1, I_2, \dots, I_n]$ with its mean \bar{I}_A and SVD $A - \bar{I}_A = U \Sigma V^T$. We also have another matrix $B = [I_{n+1}, I_{n+2}, \dots, I_{n+m}]$, which has m new target observations and its mean is \bar{I}_B . Let $C = [A, B]$ and we can compute \bar{I}_C as well as U' and Σ' by the following steps in Algorithm 1. With this incremental updating scheme, we can get the new eigen template set U' .

Algorithm 1: Incremental updating scheme for eigen template set

Step 1 Compute the mean vector $\bar{I}_C = \frac{fn}{fn+m} \bar{I}_A + \frac{m}{fn+m} \bar{I}_B$ (f is the forgetting factor).

Step 2 Form the matrix $\hat{B} = [(I_{n+1} - \bar{I}_B) \dots (I_{n+m} - \bar{I}_B)] \sqrt{\frac{nm}{n+m}} (\bar{I}_B - \bar{I}_A)$.

Step 3 Compute $\tilde{B} = orth(\hat{B} - U U^T \hat{B})$ and $R = \begin{bmatrix} \Sigma & U^T \hat{B} \\ 0 & \tilde{B}(\hat{B} - U U^T \hat{B}) \end{bmatrix}$. Where, $orth(\cdot)$

performs orthogonalization.

Step 4 Compute the SVD of R : $R = \tilde{U} \tilde{\Sigma} \tilde{V}'$.

Step 5 $U' = [U \quad \tilde{B}] \tilde{U}$ and $\Sigma' = \tilde{\Sigma}$

3.4. Algorithm Summary

The proposed tracking algorithm is summarized in Algorithm 2. The target in the first frame is obtained manually. Targets in the following partial sequence are obtained by kd-tree method and are used to generate eigen template set for initialization. Then, we formulate the tracking process as the particle filter framework. Once the target gotten in the current frame is not occluded, the incremental PCA learning steps will be used to update the eigen template set to handle the appearance variation of the target. This tracking process will be continued until the end of the video.

Algorithm2: Visual Tracking with Fragments-Based PCA Sparse Representation

Input: The initial target state $X_0 = [x_0, y_0, s_0, \theta_0, \alpha_0, \phi_0]$

1. Initialization: Construct image set $A = [I_1, I_2, \dots, I_n]$ and generate eigen template set U
 2. Sample the target area according to the motion model (12).
 3. Calculate the fragments-based PCA sparse representation coefficient of each sample and obtain the likelihood under the observation model (13).
-

-
4. Estimate the current state x_t by MAP according to the particle filter framework and get the target image y_t .
 5. Using equation (14) to judge whether the target image y_t is occluded or not. If not, using the incremental PCA learning steps to update the eigen template set.
 6. Go to step 2 until the last frame of the video is gotten.

Output : The current state x_t of target

4. Experimental Results

In this section, we present experiments to demonstrate the effectiveness of our algorithm. Six public video sequences are used for evaluating the performance of our tracker under challenges of significant occlusion, pose and illumination changes. The details of the selected video sequences are listed in Table 1. All the experiments are conducted using a MATLAB implementation on a 3GHz machine with 3GB RAM. We also test the L1 tracker [8], IVT tracker [7] and SLSAM tracker [9] for comparison. Each tracker runs with its default parameters. In our method, we partition the observed samples and eigen templates into ten local areas. We also use 30 eigen templates and 600 particles.

Table 1. Videos used in the Experiment

| Video | Frame | Main challenge |
|-----------------------|-------|--|
| David[9] | 462 | Moving camera, illumination variation |
| Car11[9] | 393 | Illumination variation, scale change, background clutter |
| Board[14] | 461 | 3D motion, appearance change |
| Faceocc2[15] | 819 | Occlusions, heavy appearance change |
| Sylvester[15] | 900 | 3D motion, illumination change |
| ThreePastShop2cor[16] | 351 | Occlusions, scale change |

4.1. Quantity Evaluation

The ground truths of the videos are provided by reference [9] and reference [16]. Three criteria are employed to evaluate the performance of the trackers. Figure 3 presents the tracking error (in pixels), which measures the Euclidean distance between the center of the ground truth and the tracking window. Table 2 summarizes the average tracking errors for each video and shows that our method has the lowest average tracking error for all the six videos. Table 3 is the accurate rate. The accurate rate indicates the percentage of the tracking success frame. We consider the tracking is a success when the distance of the tracking window center and the ground truth is less than half of the diagonal length of the rectangle enclosing the true target. From the results, we can see that our method outperforms other three state-of-the-art tracking methods and has the highest tracking accurate rate.

Table 2. Mean Distance of the Tracking Result to Ground Truth

| | David | Car11 | Sylvester | Faceocc2 | ThreePastShop2cor | Board |
|------------|---------------|---------------|----------------|---------------|-------------------|----------------|
| L1 | 20.6975 | 8.7354 | 78.9034 | 64.4524 | 63.6194 | 158.1386 |
| IVT | 7.5955 | 19.7774 | 82.8920 | 27.8803 | 43.8268 | 192.7382 |
| SLSAM | 7.8096 | 2.0672 | 51.1517 | 4.9336 | 20.8740 | 12.7683 |
| Our method | 3.4879 | 1.9522 | 18.7078 | 4.2949 | 19.2247 | 12.3096 |

Table 3. Percentage of Frames Tracked Correctly

| | David | Car11 | Sylvester | Faceocc2 | ThreePastShop2cor | Board |
|------------|------------|------------|------------|-------------|-------------------|------------|
| L1 | 31% | 70% | 12% | 32% | 23% | 24% |
| IVT | 93% | 74% | 32% | 87% | 22% | 17% |
| SLSAM | 96% | 72% | 40% | 92% | 78% | 93% |
| Our method | 96% | 88% | 73% | 100% | 85% | 98% |

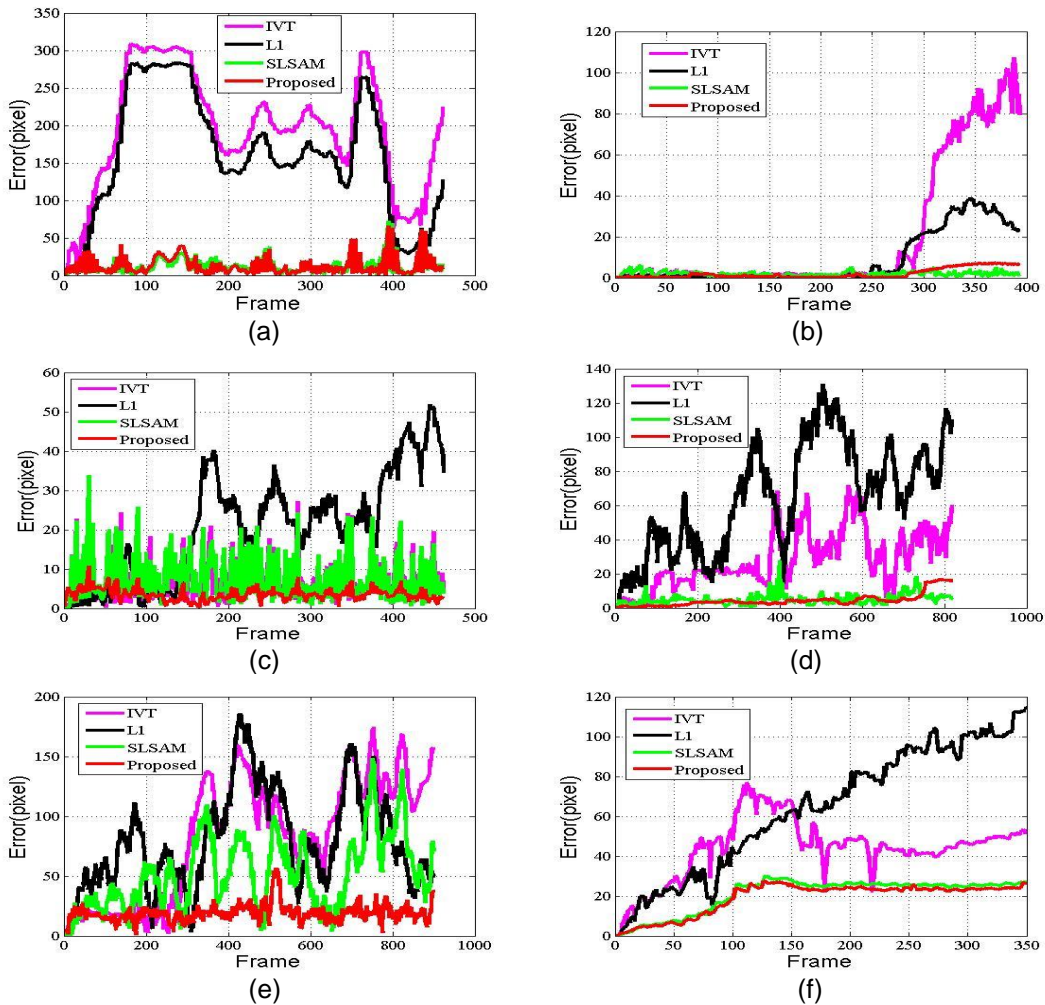


Figure 3. Tracking Errors Comparison of the Six Videos: (a) Board, (b)Car11, (c)David, (d) Faceocc2, (e) Sylvester, (f) ThreePastShop2cor

4.2. Quality Evaluation

Figure 4 shows the tracking results in Faceocc2 sequence and ThreePastShop2cor sequence, which demonstrate that fragments-based PCA sparse representation can help avoid much influence of occlusion and better estimate the target. In the Faceocc2 sequence, our method can track the face accurately even when it is heavily occluded by the book and the cap, while other trackers drift away from the target or scale badly. The L1 tracker shows good results in the first 250 frames, but then drifts away and fails to locate the face at frame 551. The IVT tracker also drifts away from frame 551 when long duration occlusion occurs. The SLSAM tracker performs competitively in this video because of the structural local sparse appearance model. However, it does not scale well when the face is heavily occluded. In the ThreePastShop2cor sequence, other three methods fail to track the target because there are two other men around and the man is two times occluded by the two men. Our tracker does not drift away and is not affected much by occlusion owing to the fragments-based PCA sparse representation and eigen template update scheme.

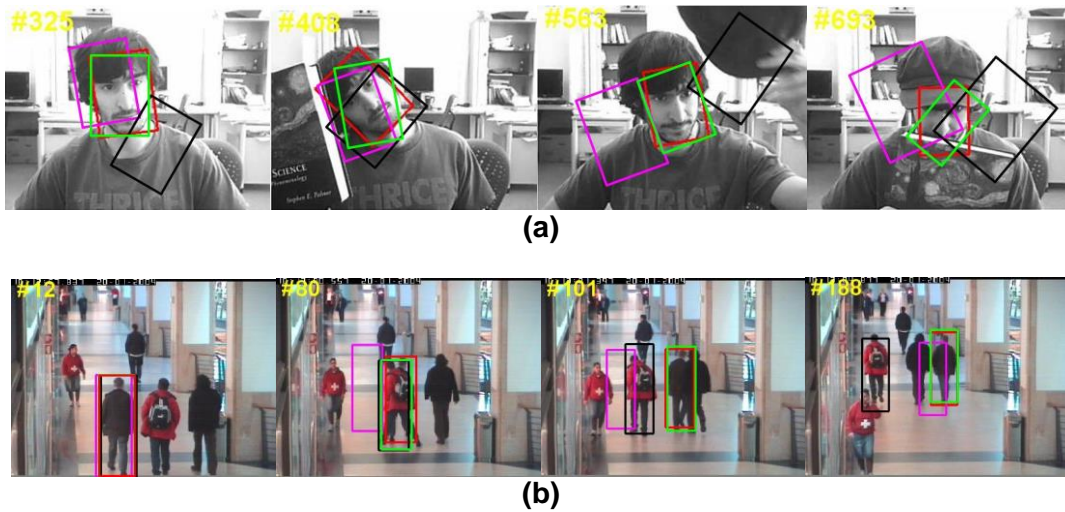


Figure 4. Tracking Results of Faceocc2 Sequence and ThreePastShop2cor Sequence: (a) Tracking Results of Faceocc2 Sequence, (b) Tracking Results of ThreePastShop2cor Sequence. The Red Rectangle Denotes our Method, the Green one is SLSAM Tracker, the Pink is the IVT Tracker and the Black Rectangle is L1 tracker

Figure 5 presents the tracking results in David sequence and Car 11 sequence with large illumination variation. In the Car 11 sequence, there is low contrast between the target and the background. Our method performs well in tracking the vehicle while the IVT tracker and L1 tracker drift to the cluttered background when drastic illumination variation occurs. Although, the SLSAM tracker keeps tracking the target to the end, our algorithm achieves low tracking error and high success rate. This can be attributed to the use of fragments-based PCA sparse representation and the incremental subspace learning which are able to handle appearance change due to lighting change. In the David sequence, the boy walks out of the dark room and into an area with spot lights. The L1 tracker does the worst and loses the target from frame 150. Our tracker can track the boy successfully. The IVT tracker and SLSAM tracker are also able to follow the boy in most of the video but with higher tracking error.



Figure 5. Tracking Results of Car 11 Sequence and David Sequence: (a) Tracking Results of Car 11 Sequence, (b) Tracking Results of David Sequence. The Red Rectangle Denotes our Method, the Green One is SLSAM Tracker, the Pink is the IVT Tracker and the Black Rectangle is L1 Tracker

Figure 6 presents the tracking results in the Board sequence and Sylvester sequence with large appearance change. For Board sequence, IVT tracker and L1 tracker drift away from the target as they are not effective in handling objects with large shape variations. The SLSAM tracker and proposed methods are able to track the target better due to the use of fragment-based appearance models. The Sylvester sequence is challenging as there are large pose variations. Our method successfully keeps tracking the target throughout the sequence. Failures are observed from frame 403 for IVT tracker, frame 134 for L1 tracker and frame 397 for SLSAM tracker.

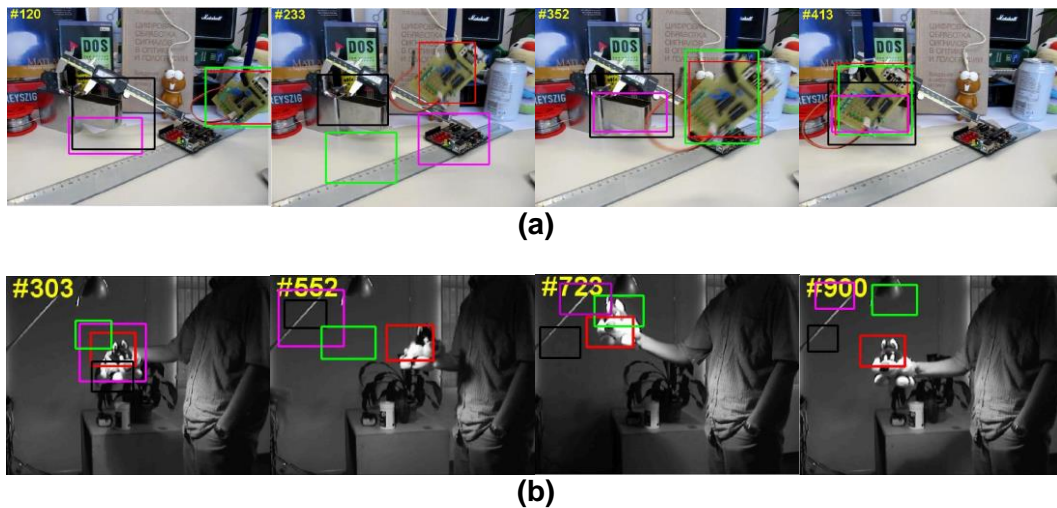


Figure 6. Tracking Results of Board Sequence and Sylvester Sequence: (a) Tracking Results of Board Sequence, (b) Tracking Results of Sylvester Sequence. The Red Rectangle Denotes our Method, the Green One is SLSAM Tracker, the Pink is the IVT Tracker and the Black Rectangle is L1 Tracker

5. Conclusion

In this paper, we propose a novel robust appearance model based on fragments-based PCA sparse representation for visual tracking, which exploits the strength of both subspace learning and sparse representation for modeling object appearance. By exploits the advantage of subspace representation, the tracking method can handle the undergoing pose and illumination changes for visual tracking. In addition, by introducing fragments-based method to sparse representation, our method exploits the spatial information as well as the partial information, which helps it more effective in handling occlusion. Finally, experimental results on challenging videos show that our method can track accurately and robustly, and outperform many other state-of-the-art trackers.

References

- [1] A. Yilmaz, O. Javed and M. Shah, "Object tracking: A survey", *ACM Computing Surveys*, vol. 38, no. 4, (2006), pp.1-45
- [2] I. Su Kim, H. Seok Choi, K. Moo Yi, J. Young 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.
- [3] V. Jelača, A. Pižurica, J. Oswaldo 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
- [4] 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.
- [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] W. Dong, F. Chang and T. Li, "Adaptive Fragments-based Target Tracking Method Fusing Color Histogram and SIFT Features", *Journal of Electronics and Information Technology*, vol. 35, no. 4, (2013), pp. 770-776.
- [7] D. A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, "Incremental learning for robust visual tracking", *International Journal of Computer Vision*, vol. 77, no. 3, (2008), pp. 125-141.
- [8] 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.
- [9] 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.
- [10] X. Zhang, W. Hu, S. Maybank and X. Li, "Graph based discriminative learning for robust and efficient object tracking", *IEEE 11th International Conference on Computer Vision*, (2007), pp. 1-8.
- [11] Y. Zha, Y. Yang and D. Bi, "Graph-based transductive learning for robust visual tracking", *Pattern Recognition*, vol. 43, no. 1, (2010), pp. 187-196.
- [12] M. Yang, Z. Fan, J. Fan and Y. Wu, "Tracking non-stationary visual appearances by data-driven adaptation", *IEEE Transactions on Image Processing*, vol. 18, no. 7, (2009), pp. 1633-1644.
- [13] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, "RobustFace Recognition via Sparse Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, (2009), pp. 210-227.
- [14] J. Santner, C. Leistner, A. Saffari, T. Pock and H. Bischof, "PROST: Parallel robust online simple tracking", *IEEE Conference on Computer Vision and Pattern Recognition*, (2010), pp. 723-730.
- [15] 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.
- [16] CAVIAR [Online]. Available: <http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>.

Author



Peishu Qu, he received his B.S. degree in electronic engineering from Qufu Normal University in 2003 and M.S. degree in communication and information systems from Civil Aviation University of China in 2009. He focuses on computer vision, pattern recognition and image processing.