Object Tracking via Pixel-Wise and Block-Wise Sparse Representation

Pouria Navaei, Mohammad Eslami(✉), and Farah Torkamani-Azar

Cognitive Telecommunications Research Group,
Department of Electrical Engineering,
Shahid Beheshti University, Evin, 1983963113 Tehran, Iran
{p_navaei,m_eslami,f-torkamani}@sbu.ac.ir

Abstract. Object tracking is an important task within the field of computer vision. In this paper, a new and robust method for target tracking in video sequences is proposed based on sparsity representation. Also, in order to increase the accuracy of the tracking, the proposed method uses both group and individual sparse representations. The appearance changes of the target are considered by an on-line subspace training and the appearance model is updated in a procedure which is modified by considering both global and local analysis which brings more accurate appearance model. The proposed appearance representation model is exploited along with the particle filter framework to estimate the target's state and our particle filter uses a modified observation model too. This method is evaluated on several tracking benchmark videos with some different tracking challenges. The results show the robustness of the proposed method in dealing with challenges such as occlusions, changes in illuminations and poses with respect to other related methods.

Keywords: Appearance representation model · Target tracking · Sparse representation · Particle filter

1 Introduction

Target tracking in videos or visual tracking plays a key role in many fields of computer vision applications such as intelligent surveillance, intelligent transportation, activity recognition and etc. Although many algorithms have been proposed, but still some challenges have remained in which researchers have interest to solve them. Most of the visual tracking algorithms consist of three components

- *Motion model:* is used to predict the state of the target in the frame.
- *Appearance model:* represents the appearance of the target according to its visual characteristics.
- *Search method:* considers the appearance and motion models to select the most likely target's state.

P. Sharma and F.M. Bianchi (Eds.): SCIA 2017, Part I, LNCS 10269, pp. 273–284, 2017.
DOI: 10.1007/978-3-319-59126-1_23

The main challenge in designing a robust tracking algorithm is changes in the appearance of the target caused by blurring, non-uniform illuminating, size changing and partial occlusions. Therefore, appearance model is the one of the key components in the robust tracking that has received more attention in the recent years [1].

In the most of previous tracking methods, appearance model was based on the templates [1–3] or subspaces [4]. However, these methods are not suitable when occlusions or drastic changes occur in the target appearance. Recently, a some of appearance model techniques is presented based on the sparse representation, which have more desirable performance in dealing with appearance corruptions and especially occlusions [5–9]. Sparsity representation has many attractive applications such as compressive sensing, dimension reduction, source separation, super resolution [10] in computer vision and also in other subjects of signal processing such as classification [11], cognitive radios [12] and etc. First of sparse representation based tracking method was proposed in [5] by Mei and Ling, which has some unsolved problems such as high computational cost, low number of templates in the dictionary and occlusion effects in the updated dictionary. Therefore further efforts were done to solve these problems e.g. articles [6,7] have been able to address the problems of the paper [5]. However, as the results of experiments show both methods have low accuracy in some scenarios yet. In this paper inspired from [6,7], an effective tracking method is proposed based on both block and pixel based sparsity representations which its results represent the more accuracy and tracking stability with them.

The rest of this paper is organized as follow. The sparse tracking method is described in Sect. 2 and the proposed method is suggested in Sect. 3. Section 4 presents the experimental results and finally the paper is concluded in Sect. 5.

2 Tracking Based on Sparse Representation

In this section, the basics of tracking based on sparsity representation is introduced. In addition, two relevant recent well cited methods called as Sparse Prototypes Tracker (SPT) [6] and Structured Sparse Representation Tracker (SSRT) [7] are discussed too.

2.1 Original Sparse Representation Model

Mei and Ling proposed sparsity representation based tracking method [5]. In this type, target appearance is modeled by a sparse linear combination of target and trivial templates as shown in Fig. 1. In fact, they propose an algorithm (l_1 tracker) by casting the tracking problem as finding the most likely patch with sparse representation and handling partial occlusion with trivial templates. Trivial templates is an identity matrix and is exploited to model occlusion and noise in the real-world observation data. More precisely $\boldsymbol{y} \in R^d$ could be the observation vector as:

$$\boldsymbol{y} \cong \boldsymbol{T}\boldsymbol{a} + \boldsymbol{e} = \begin{bmatrix} \boldsymbol{T}\ \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{e} \end{bmatrix} = \boldsymbol{D}\boldsymbol{c} \tag{1}$$

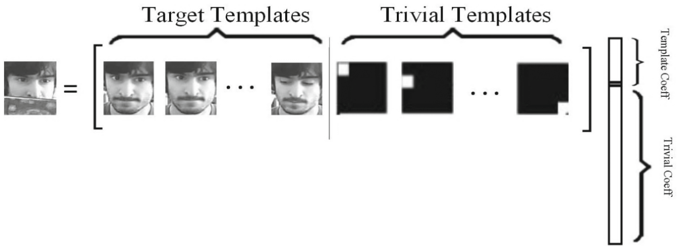

Fig. 1. Original sparse representation model for target tracking [5].

where $T = [\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_m] \in R^{d\times m}$ $(d \gg m)$ is the set of training templates and $I \in R^{d\times d}$ is the trivial templates, which $\left[\boldsymbol{T}\ \boldsymbol{I}\right]$ can be assumed as a dictionary of representation. Vector $\boldsymbol{a} \in R^m$ is the coefficients vector and $\boldsymbol{e} \in R^d$ is the error vector in which indicates the partial occlusion. The occlusion only covers a portion of the target appearance and therefore it is possible to assume that the error vector $\boldsymbol{e}$ and consequently vector $\boldsymbol{c} \in R^{d+m}$ are sparse [5]. To find the sparse vector $\boldsymbol{c}$, the following minimization problem should be solved,

$$\min_{\boldsymbol{c}} \frac{1}{2} \left\|\boldsymbol{Dc} - \boldsymbol{y}\right\|_2^2 + \lambda \left\|\boldsymbol{c}\right\|_1 \tag{2}$$

where $\|\|_2$ and $\|\|_1$ denote norms l_2 and l_1 respectively.

2.2 Sparse Prototypes Tracker (SPT)

In article [6], Wang et al. proposed an extension named as *Sparse Prototypes Tracker (SPT)* for target representing. They exploit the strength of both subspace learning and sparse representation for modeling object appearance. For object tracking, they model target appearance with PCA basis vectors $\boldsymbol{U}$, and account for occlusion with trivial templates $\boldsymbol{I}$ by

$$\boldsymbol{y} \cong \boldsymbol{Uz} + \boldsymbol{e} = \left[\boldsymbol{U}\ \boldsymbol{I}\right] \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{e} \end{bmatrix} \tag{3}$$

where $\boldsymbol{z}$ indicates the coefficients of basis vectors. In their formulation, the prototypes consist of just a small number of PCA basis vectors, therefore the $\boldsymbol{z}$ will be dense and the appearance problem can be modified as follow. Figure 2 shows the difference in representations of [5] and SPT [6] which target templates are replaced by PCA basis. Prototypes consist of PCA basis vectors and trivial templates.

$$\min_{\boldsymbol{z},\boldsymbol{e}} \frac{1}{2} \left\|\boldsymbol{y} - \boldsymbol{Uz} - \boldsymbol{e}\right\|_2^2 + \lambda \left\|\boldsymbol{e}\right\|_1 \tag{4}$$

It is obvious that the number of used basis vectors in matrix $\boldsymbol{U}$ could be effective on accuracy.

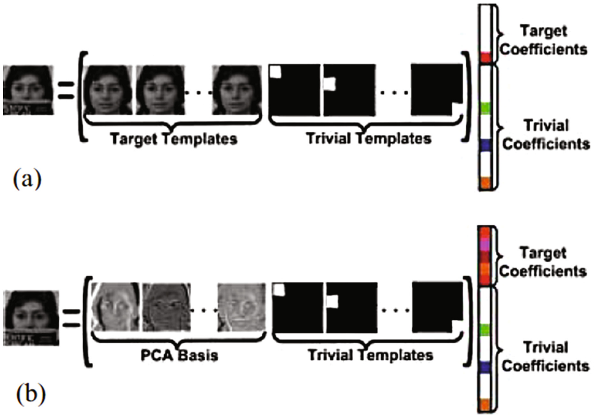

Fig. 2. Sparse representation models for target tracking. (a) Original [5] (b) Sparse Prototypes Tracker (SPT) [6].

2.3 Structured Sparse Representation Tracker (SSRT)

In SPT [6], authors only use information from individual pixels and do not exploits any predetermined assumptions about the structure of the sparse coefficients. But the performance of using the group sparsity or structured sparsity is higher than using just original sparsity [9]. In other words, having previous knowledge of the signal's structure and exploiting it can yield the better results. *Structured Sparse Representation Tracker (SSRT)* is proposed in [7] by Bai and Li with assuming continuous occlusion and previous knowledge of the dictionary structure. As shown in Fig. 3, authors first partition the observed sample and also each of the training templates into R local parts which makes contiguous occlusion (highlighted with red) can be stacked (grouped) as a block sparse vector that has clustered nonzero entries. Then the partitioned regions are stacked into $1-D$ vectors $\boldsymbol{y}$. Also, Corresponding structuring should be considered for PCA or subspace templates. More details can be found in [7].

3 Proposed Tracking Algorithm

In this section, our proposed method based on SPT and SSRT methods is explained. The proposed appearance model is defined first and then the particle filter tracking framework is adjusted for coping the model. Finally, procedure for updating the appearance model is discussed.

3.1 Proposed Appearance Model

Based on tracking with structured sparse representation model, since the occlusion geometry is unknown, therefore regardless of the occlusion geometry, the sample is partitioned into predefined blocks. In cases where occlusion does not

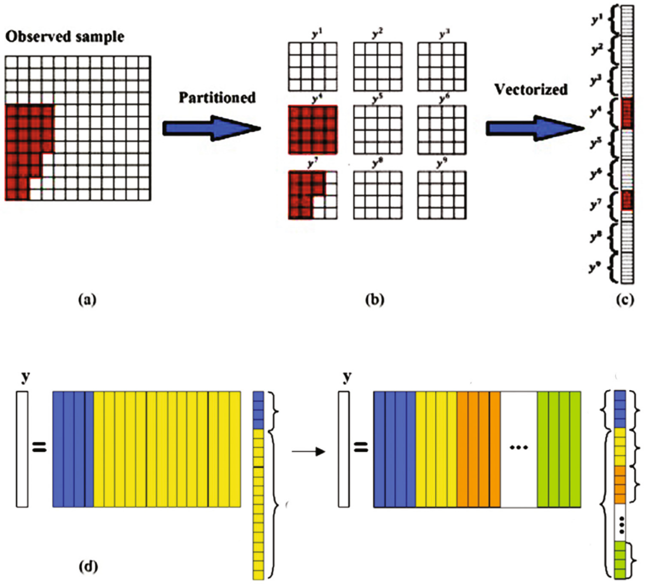

Fig. 3. A simple illustration of structured sparse representation. (a) Observed holistic sample or template, (b) Partition the sample into the local areas, (c) Convert local areas into vectors and putting them in an observed vector, (d) Block structured basis. (Color figure online)

completely fill a block (e.g. 7th block in Fig. 3), the block may be determined as clean (without occlusion) or in contrast full of occlusion, and this simple decision criterion leads to a weak accuracy in tracking procedure.

In order to solve this problem, we propose to represent the appearance model of the target by using original sparse representation of pixels and group sparse representation simultaneously. In this model, $l_{2,1}$ and l_1 norms are used to represent group and individual pixel sparsity, respectively. The proposed sparse tracking model is:

$$\min_{\boldsymbol{z},\boldsymbol{e}} \frac{1}{2} \left\| \bar{y} - \boldsymbol{U}\boldsymbol{z} - \boldsymbol{e} \right\|_2^2 + \lambda_1 \left\| \boldsymbol{e} \right\|_1 + \lambda_2 \left\| \boldsymbol{e} \right\|_{2,1} \tag{5}$$

where $\boldsymbol{U}$ is the PCA subspace extracted from target templates. Also, $\boldsymbol{e}$ is the error vector that includes $\boldsymbol{e} = \left[e^{1^T}, e^{2^T}, \cdots, e^{J^T} \right]$ where J is the total number of blocks and $\boldsymbol{e}^j$ is the error vector for the jth block. In this fashion, the lower size of data can model more states of the object. The vector $\bar{y}$ is the centered observation vector, i.e. $\bar{y} = \boldsymbol{y} - \boldsymbol{\mu}$ which $\boldsymbol{\mu}$ is the average vector of the training space. The subspace coefficients $\boldsymbol{z}$ and sparse error vector $\boldsymbol{e}$ should be found while vector $\boldsymbol{e}$ is considered regarded to both pixel based and group based sparseness properties. Pixel based sparseness of the error vector is considered by $\left\| \boldsymbol{e} \right\|_1$. The sparseness in groups is computed by

$$\|\boldsymbol{e}\|_{2,1} = \sum_{j=1}^{J} \left\|\boldsymbol{e}^j\right\|_2. \tag{6}$$

The coefficients of λ_1 and λ_2 control the sparseness of the pixel based and groups based sparseness.

3.2 Particle Filter

For robust tracking, we exploit the proposed appearance model in the particle filter tracking framework and estimate the state of the target's [13]. The motion model in the particle filter is modeled by a Gaussian distribution around the target's state in the previous frame. This means

$$p(\boldsymbol{x}_t|x_{t-1}) = \mathcal{N}(\boldsymbol{x}_t : \boldsymbol{x}_{t-1}, \Psi) \tag{7}$$

where $\boldsymbol{x}_t$ is the target's state vector at tth frame and Ψ is the covariance matrix of the target's states. The state vector $\boldsymbol{x}_t = (x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t)$ contains six parameters as state variables where $x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t$ denote x, y translations, rotation angle, scale, aspect ratio, and skew respectively. Observation likelihood function is calculated as:

$$p(\boldsymbol{y}_t|x_t) = \exp(-\left\|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t\right\|_2^2) \tag{8}$$

where $\hat{\boldsymbol{y}}_t$ is prediction of the observed sample in the tth frame based on state $\boldsymbol{x}_t$. The formula $\hat{\boldsymbol{y}}_t = \boldsymbol{Ta}$ is used in literature of tracking for particle filtering. However, we propose a modified observation model which is inspired by [6] as follow.

$$\begin{aligned} p(\boldsymbol{y}_t|x_t) &= \exp(-\left[\|\boldsymbol{y}_t - \boldsymbol{Uz} - \boldsymbol{e}\|_2^2 + \lambda_2 \times NOEB\right]) \\ &= \exp(-\left[term1 + term2\right]) \end{aligned} \tag{9}$$

As mentioned before, similar criterion is proposed in the SPT method of [6], but their reconstruction error ($term1$) was just calculated over the pixels without occlusions. In Eq. (9), we also consider the number of occlusion blocks in $term2$ as $NOEB$, which is the sum of the *Error Number* of each block in an observed sample. Figure 3 shows the concept of the observed sample which contains some blocks. Suppose that, the number of blocks in each observed sample is J, then the $NOEB$ is $NOEB = \sum_{j=1}^{J} \gamma_j$ where γ_j is the *Error Number* of each block and is computed as follow.

$$\gamma_n = \frac{number\ of\ occluded\ pixels\ in\ the\ nth\ block}{number\ of\ pixels\ in\ the\ nth\ block} \tag{10}$$

In addition, two thresholds tr_L and tr_H are used to define three types of Error Number as follow.

- If $\gamma_j \leq tr_L$, the block is considered as error-free and Error Number will be set to $\gamma_j = 0$.
- If $\gamma_j \geq tr_H$, the block is considered as completely error block and Error Number will be set to $\gamma_j = 1$.
- If $tr_L \geq \gamma_j \leq tr_H$, some of the pixels in the block have errors and Error Number will be set to γ_j.

3.3 Updating Appearance Model

Because of changes in the appearance of the targets during tracking sequences, it is not logical to use a fixed subspace for target appearance representation. Therefore updating the appearance model dynamically could improve the tracking performance. It is important to update by using the *correct templates* which has no errors such as occlusion and background. So the first step of the updating procedure is to select the *correct templates*. We propose to use a local analysis along with global analysis. Suppose that observed sample $\boldsymbol{y}$ is selected by the particle filter and the corresponding error vector is computed as $\boldsymbol{e}$ by (5).

In global analysis if the number of occlusion blocks in the selected error vector $\boldsymbol{e}$, is greater than a certain threshold, then the sample $\boldsymbol{y}$ will be rejected and not be used for updating. Otherwise this sample will be used to update the subspace after local analysis as follows. For each block j:

- If $\gamma_j \leq tr_L$, the entire block pixels remains unchanged.
- If $\gamma_j \geq tr_H$, the entire block pixels are replaced with the mean vector of the subspace $\boldsymbol{\mu}$.
- If $tr_L \geq \gamma_j \leq tr_H$, only occlusion pixels are replaced by corresponding values in the mean vector $\boldsymbol{\mu}$ and other pixels left unchanged.

After determining the *correct templates* based on above mentioned procedure and collects them (e.g. 5 corrected templates), they will be used to update the subspace $\boldsymbol{U}$ and the mean vector $\boldsymbol{\mu}$ by exploiting the incremental learning algorithm presented in [4].

4 Experimental Results

The proposed tracking algorithm was simulated in the Matlab platform while CVX is used to solve (5) [14,15]. In order to evaluate the performance of the proposed algorithm, four different sequences are selected in which have different tracking challenges as shown in Table 1.

The results of the proposed algorithm are compared with two other sparse tracking algorithms, SPT-2013 [6] and SSRT-2012 [7]. While the SPT simulation codes have been written by its author and are available to use, the simulation code for SSRT algorithm are provided by ourselves.

Table 1. Dataset characteristics: length and challenges

Dataset	Frames	Challenges
David	1–470	Pose and illumination variation, occlusion
Faceocc2	1–819	In-plane rotation and occlusion
Car6	1–705	Heavy occlusion
Jumping	1–313	Fast motion and motion Blur

Each observed sample is resized to size 32 × 32 for SPT and our method and 15 × 12 for SSRT. Each observed sample is partitioned to 64 and 6 blocks for our method and SSRT, respectively. The number of 600 particles are selected for particle filter. In all experiments, $\lambda_1 = 0.02$, $\lambda_2 = 0.27$, $tr_L = 0.25$ and $tr_H = 0.75$. The number of basis for PCA subspace is 10.

In order to evaluate and compare the proposed method with other algorithms quantitatively, the overlapping diagrams are drawn. *Overlap Rate (OLR)* [16,17], the overlap area between the detected target and the area specified by the ground truth is defined as

$$OLR = \frac{area\,(ROI_{TR} \cap ROI_{GT})}{area\,(ROI_{TR} \cup ROI_{GT})} \tag{11}$$

where the ROI_{TR} is the target's ROI which is the result of the tracking algorithm and ROI_{GT} is the corresponding correct area in the Ground truth. In addition, the *Center Location Error (CLE)*, the Euclidean distance between the centers of the found target and Ground truth is computed.

Figure 4 shows the results of the "David" sequence which contains both changing in illumination and target state. The Overlap Rate diagrams are illustrated in Fig. 4(b) and show that the performance of the proposed algorithm is better than other algorithms.

Figure 5 shows the results of the "faceocc2" sequence. In this sequence partial occlusions occur along with rotation of the target. The performance of the proposed algorithm is same as other algorithms until the 700th frame. But after a large occlusion (700th frame), the proposed algorithm brings much better performance than the two other algorithms.

Figure 6 is about sequence of the "car6" where the vehicle moves and a large occlusion occurs at 280th frame. While SPT and SSRT fail to track, the proposed algorithm tracks the target in all frames of this sequence as well!

Figure 7 shows the results of the "jumping" sequence. The target have fast motion and blurred in the most of the frames. The Overlap Rate diagrams are illustrated in Fig. 7(b) and show that the performance of different trackers. It can be inferred from Fig. 7 that, while the proposed algorithm performs almost similar to SPT, but is very better than SSRT algorithm.

Furthermore, the quantity measurements are reported here for each sequence and algorithm. The average of CLE and OLR of the algorithms for all frames are

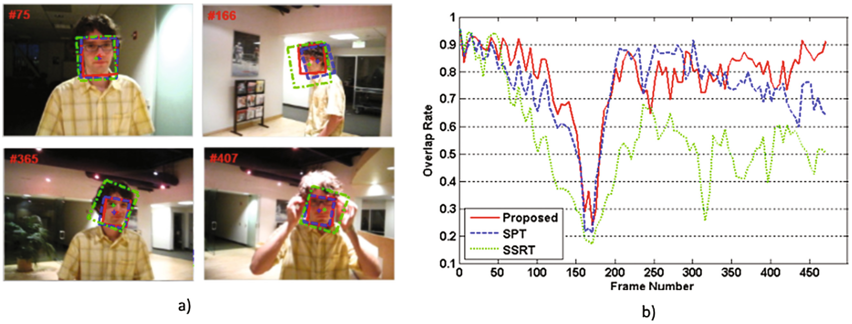

Fig. 4. Tracking results of the proposed tracker, SPT tracker and SRRT tracker on "David" sequence. (a) Quality evaluation, (b) Overlapping rate diagram.

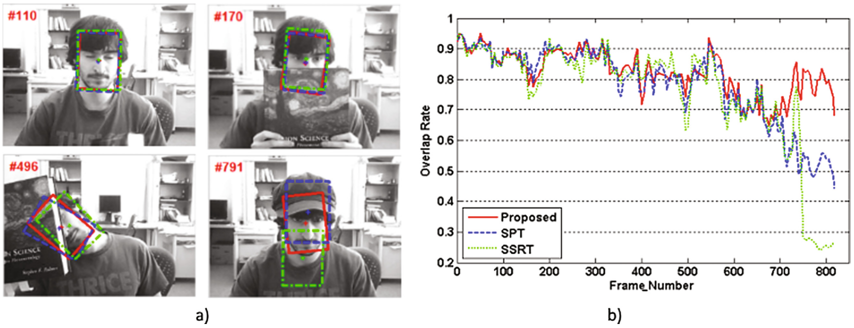

Fig. 5. Tracking results of the proposed tracker, SPT tracker and SRRT tracker on "Faceocc2" sequence. (a) Quality evaluation, (b) Overlapping rate diagram.

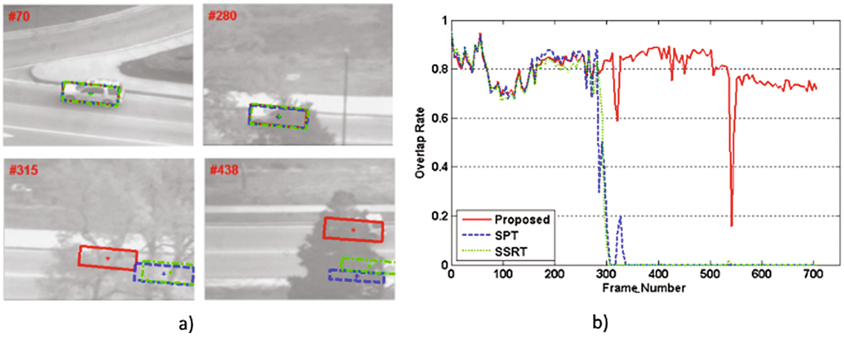

Fig. 6. Tracking results of the proposed tracker, SPT tracker and SRRT tracker on "Car6" sequence. (a) Quality evaluation, (b) Overlapping rate diagram.

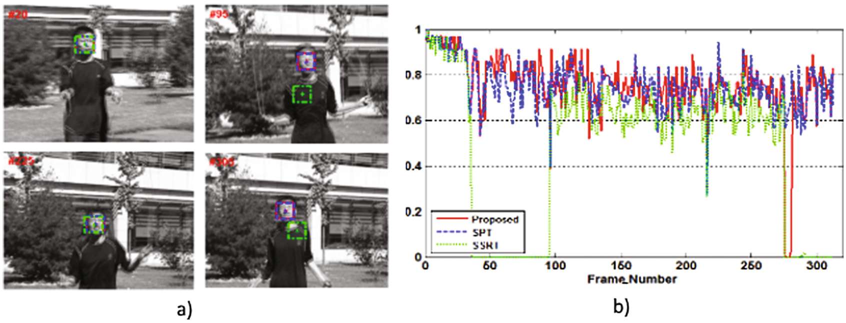

Fig. 7. Tracking results of the proposed tracker, SPT tracker and SRRT tracker on "jumping" sequence. (a) Quality evaluation, (b) Overlapping rate diagram.

reported in the Tables 2 and 3 for each sequence. As shown in the table, the lowest average CLE and maximum OLR are denoted with bold notation. Results show that, the proposed algorithm and criteria brings better performance of sparsity based tracking.

Finally, in this paper, new criteria are proposed in (5) and (9) to represent and track objects more precisely and robustly. But, for practical usage, it is necessary to deal with real-time sequences. Therefore, we try to extended this work and propose a new and fast solution algorithm in future. On the other hand, using just one adaptive subspace (PCA based) to represent the objects is not comprehensive and it is better to train more different or nonlinear spaces to improve the representation capability. Also, extracting the background infor-

Table 2. Average center location error (CLE) of different methods for considered videos.

Videos	Proposed	SPT [6]	SSRT [7]
David	**3.24**	4.25	5.52
Faceocc2	**4.24**	6.17	8.85
Car6	**3.46**	78.01	86.10
Jumping	4.0	**3.69**	22.04

Table 3. Average overlap rate (OLR) of different methods for considered videos.

Videos	Proposed	SPT [6]	SSRT [7]
David	**0**.78	0.74	0.55
Faceocc2	**0**.82	0.78	0.76
Car6	**0**.79	0.34	0.33
Jumping	**0**.78	0.75	0.46

mation of the image and considering it can be useful to model and find the new representation spaces.

5 Conclusion

This paper proposed a robust and fast tracking algorithm using sparse representation along with particle filtering. In order to represent the target's appearance, simultaneous pixel based and block based sparse representations are considered. Based on blocking and grouping concepts which are used to develop the appearance model, a new observation model in particle filter is suggested too. Finally, a simple additional criterion is proposed to select and modify the correct templates which will be used for PCA subspace updating. Experiments show the robustness of the proposed tracking algorithm according to major challenges such as occlusion, illumination changes, resizing, rotating and also brings better performance in comparison with recent SPT and SSRT algorithms.

References

1. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., Hengel, A.V.D.: A survey of appearance models in visual object tracking. ACM Trans. Intell. Syst. Technol. (TIST) **4**, 58 (2013)
2. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 564–577 (2003)
3. Chateau, T., Laprest, J.T.: Realtime Kernel based tracking. ELCVIA Electron. Lett. Comput. Vis. Image Anal. **8**(1), 27–43 (2009)
4. Ross, D.A., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for Robust visual tracking. Int. J. Comput. Vis. **77**, 125–141 (2008)
5. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: IEEE 12th International Conference on Computer Vision, pp. 1436–1443 (2009)
6. Wang, D., Lu, H., Yang, M.-H.: Online object tracking with sparse prototypes. IEEE Trans. Image Process. **22**, 314–325 (2013)
7. Bai, T., Li, Y.: Robust visual tracking with structured sparse representation appearance model. Pattern Recogn. **45**, 2390–2404 (2012)
8. Zhuang, B., et al.: Visual tracking via discriminative sparse similarity map. IEEE Trans. Image Process. **23**(4), 1872–1881 (2014)
9. Huang, J., Zhang, T.: The benefit of group sparsity. Ann. Stat. **38**, 1978–2004 (2010)
10. Baraniuk, R.G., et al.: Applications of sparse representation and compressive sensing [scanning the issue]. In: Proceedings of the IEEE, vol. 98, no. 6, pp. 906–909 (2010)
11. Zhang, L., Yang, M., Feng, X.: Sparse representation or collaborative representation: which helps face recognition? In: IEEE International Conference on Computer Vision (ICCV) (2011)
12. Eslami, M., Torkamani-Azar, F., Mehrshahi, E.: A centralized PSD map construction by distributed compressive sensing. IEEE Commun. Lett. **19**(3), 355–358 (2015)

13. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Sig. Process. **50**, 174–188 (2002)
14. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 2.0 beta, September 2013. http://cvxr.com/cvx,
15. Grant, M.C., Boyd, S.P.: Graph implementations for nonsmooth convex programs. In: Blondel, V.D., Boyd, S.P., Kimura, H. (eds.) Recent Advances in Learning and Control. LNCIS, vol. 371, pp. 95–110. Springer, London (2008)
16. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)
17. Čehovin, L., Leonardis, A., Kristan, M.: Visual object tracking performance measures revisited. IEEE Trans. Image Process. **25**(3), 1261–1274 (2016)