Visual tracking based on the sparse representation of the PCA subspace*

CHEN Dian-bing (陈典兵)[1,2]**, **ZHU Ming** (朱明)[1], **and WANG Hui-li** (王慧利)[1,2]

1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China

2. University of Chinese Academy of Sciences, Beijing 100039, China

(Received 12 April 2017; Revised 2 June 2017)

We construct a collaborative model of the sparse representation and the subspace representation. First, we represent the tracking target in the principle component analysis (PCA) subspace, and then we employ an L_1 regularization to restrict the sparsity of the residual term, an L_2 regularization term to restrict the sparsity of the representation coefficients, and an L_2 norm to restrict the distance between the reconstruction and the target. Then we implement the algorithm in the particle filter framework. Furthermore, an iterative method is presented to get the global minimum of the residual and the coefficients. Finally, an alternative template update scheme is adopted to avoid the tracking drift which is caused by the inaccurate update. In the experiment, we test the algorithm on 9 sequences, and compare the results with 5 state-of-art methods. According to the results, we can conclude that our algorithm is more robust than the other methods.

Document code: A **Article ID:** 1673-1905(2017)05-0392-5

DOI https://doi.org/10.1007/s11801-017-7080-z

Visual tracking, a theme of the computer vision, has been playing a critical role for decades, and it has been applied in numerous real-life applications, such as video surveillance, traffic analysis, human machine interfaces and robotics control[1-3].

The subspace representation[4] is an impressively efficient method in the visual tracking. The incremental visual tracking (IVT) method[5] constructs the model of the object by the principal component analysis (PCA)[6] subspace bases, and forces the bases to adapt to the appearance variation of the target through incremental update scheme. Although the subspace trackers[5,7] have effective power to deal with pose change, illumination variation and in-plane rotation, they are sensitive to some more complicated situations (e.g. partial occlusion, fast motion). The reason is that the noise term cannot be modeled with small variances[8] any more. On the other hand, sparse representation has been applied in computer vision task recently, such as face recognition[9], super-resolution[10] and background subtraction[11]. Advantage of using the sparse representation lies in the robustness to a wide range of image corruptions, especially to an occlusion[12]. Based on the theory of Ref.[9], Xue et al[13] first used the sparse representation in the visual tracking by casting the tracking target as a linear combination of the dictionary templates. However, solving the L_1 minimization usually suffers expensive computation. Xue adopted a minimum error bound. The error bound can reduce the number of the particles which are required to solve L_1 minimization[12]. Bao et al[14] used the accelerated proximal gradient (APG) method to search the optimal solution. Zhuang et al[15] employed a distance sparse similarity map to restrict the sparse coefficients and modify the APG method to obtain the coefficients in company. Wang et al[16] proposed a tracking method which depends on a novel robust linear regression and considered the error term as the Gaussian-Laplacian distribution. Xiao et al customized the L_2 regularized least square method and used it to computer the coefficients of the representation model. Compared with the L_1-based algorithm, this method provides very fast performance without the loss of accuracy in handling the tracking problem[17]. Shreyamsha Kumar et al[8] represented the tracking target linear to the orthogonal PCA base vectors and used weighted residual minimization to detect occlusion.

In this letter, we construct a collaborative model of the sparse representation and the subspace representation. We perform the algorithm in the particle filter framework. Furthermore, an iterative method is presented to solve the minimization problem. Finally, an alternative template update scheme is adopted to avoid the tracking drift. From the results of comparison, we can conclude that the proposed algorithm is more robust than the other algorithms.

* This work has been supported by the National Natural Science Foundation of China (No.61401425).

** E-mail: chendianbing1934@163.com

We implement the algorithm in the particle filter framework[13]. Assume that $\boldsymbol{x}_t$ is the state variable of the target at frame t and contains 6 affine parameters:

$$\boldsymbol{x}_t=\left(x,y,s,\theta,\alpha,\phi\right)\ , \tag{1}$$

in which (x, y, s, θ, α, ϕ) represent the x coordinate, y coordinate, scale, rotation angle, aspect of the width and height and skew, respectively. A set of image observations is given up to frame t:

$$\boldsymbol{O}_t=\left[\boldsymbol{o}_1,\boldsymbol{o}_2,...,\boldsymbol{o}_t\right]\ , \tag{2}$$

in which $\boldsymbol{o}$ corresponds to the tracking results. We can infer the posterior probability by the Bayesian theorem:

$$p\left(\boldsymbol{x}_t\left|\boldsymbol{O}_t\right.\right)\propto p\left(\boldsymbol{o}_t\left|\boldsymbol{x}_t\right.\right)\int p\left(\boldsymbol{x}_t\left|\boldsymbol{x}_{t-1}\right.\right)p\left(\boldsymbol{x}_{t-1}\left|\boldsymbol{O}_{t-1}\right.\right)\mathrm{d}\boldsymbol{x}_{t-1}\ , \tag{3}$$

where $p(\boldsymbol{x}_t|\ \boldsymbol{x}_{t-1})$ is the dynamic motion model, and $p(\boldsymbol{o}_t|\ \boldsymbol{x}_t)$ represents the observation model. Usually, the motion model follows the Gaussian distribution:

$$p\left(\boldsymbol{x}_t\left|\boldsymbol{x}_{t-1}\right.\right)=N\left(\boldsymbol{x}_t,\boldsymbol{x}_{t-1},\boldsymbol{\Sigma}\right), \tag{4}$$

where $\boldsymbol{\Sigma}$ denotes a diagonal covariance matrix and consists of the variances of the 6 affine parameters. In this letter, $\boldsymbol{y}_i$ is the image patch wrapped from the image according to $\boldsymbol{x}_i$. Then we regard $\boldsymbol{y}_i$ as the observation of the state $\boldsymbol{x}_i$. $p(\boldsymbol{y}_{i,}|\ \boldsymbol{x}_{i,})$ is inversely proportional to the scores of the candidates.

We assume that the tracking target is represented in a PCA subspace. The PCA subspace is spanned by the base set $\boldsymbol{U}\in\boldsymbol{R}^{d\times m}$ and the centre $\boldsymbol{\mu}\in\boldsymbol{R}^{d\times 1}$[16], where m is the number of the bases, d is the dimension of each base, and $\boldsymbol{\mu}$ is also the mean of the subspace. At the first frame of the tracking, we wrap the image patch according to the ground truth and define it as $\boldsymbol{\mu}$. The base set $\boldsymbol{U}$ is learned by the incremental PCA method[5]. If the image patch set of the candidates is given

$$\boldsymbol{Y}=\left[\boldsymbol{y}_1,\boldsymbol{y}_2,...,\boldsymbol{y}_n\right], \tag{5}$$

we can represent the i-th candidate by $\boldsymbol{U}$, $\boldsymbol{\mu}$ and $\boldsymbol{e}$ as

$$\overline{\boldsymbol{y}}_i=\boldsymbol{U}\boldsymbol{z}_i+\boldsymbol{e}_i\ , \tag{6}$$

where

$$\overline{\boldsymbol{y}}_i=\boldsymbol{y}_i-\boldsymbol{\mu}\ , \tag{7}$$

where $\overline{\boldsymbol{y}}_i$ indicates the image patch that contains the difference between $\boldsymbol{y}_i$ and $\boldsymbol{\mu}$, $\boldsymbol{z}_i\in\boldsymbol{R}^{m\times 1}$ indicates the representation coefficients that are related to $\boldsymbol{U}$, and $\boldsymbol{e}_i\in\boldsymbol{R}^{d\times 1}$ is the residual matrix term of the representation. Similar to $\boldsymbol{y}_i$, $\boldsymbol{e}_i$ is used in vector form. Through Eqs.(7) and (8), we can see that if $\boldsymbol{y}_i$ is close to $\boldsymbol{\mu}$, $\boldsymbol{z}_i$ and $\boldsymbol{e}_i$ should be sparse. Then we use the L_2 norm to restrict the sparsity of the coefficient $\boldsymbol{z}_i$ and the L_1 norm to restrict the residual matrix $\boldsymbol{e}_i$. Consequently, we can get the following optimization model:

$$L\left(\boldsymbol{z}_i,\boldsymbol{e}_i\right)=\frac{1}{2}\left\|\overline{\boldsymbol{y}}_i-\boldsymbol{U}\boldsymbol{z}_i-\boldsymbol{e}_i\right\|_F^2+\lambda\left\|\boldsymbol{z}_i\right\|_2^2+\beta\left\|\boldsymbol{e}_i\right\|_1, \tag{8}$$

$$\left[\hat{\boldsymbol{z}}_i,\hat{\boldsymbol{e}}_i\right]=\arg\min_{\boldsymbol{z}_i,\boldsymbol{e}_i}L\left(\boldsymbol{z}_i,\boldsymbol{e}_i\right). \tag{9}$$

As the optimization model has the L_1 regularization, it is an *NP*-hard problem and does not have a closed-form solution. However, if we fix one of the $\boldsymbol{z}_i$ and $\boldsymbol{e}_i$, the other can be computed efficiently. Thus, we present an iterative method for solving the problem[16].

(1) Fixing $\boldsymbol{e}$ and computing $\boldsymbol{z}$

The problem transforms to

$$\frac{1}{2}\left\|\overline{\boldsymbol{y}}_i-\boldsymbol{U}\boldsymbol{z}_i-\boldsymbol{e}_i\right\|_F^2+\lambda\left\|\boldsymbol{z}_i\right\|_2^2. \tag{10}$$

This is an ordinary least squares problem. The optimal solution is

$$\hat{\boldsymbol{z}}_i=\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\right)^{-1}\boldsymbol{U}^{\mathrm{T}}\frac{\overline{\boldsymbol{y}}_i-\hat{\boldsymbol{e}}_i}{1+\lambda}. \tag{11}$$

As the set of the PCA subspace base, $\boldsymbol{U}$ is orthogonal. Consequently, we can get

$$\hat{\boldsymbol{z}}_i=\boldsymbol{U}^{\mathrm{T}}\frac{\overline{\boldsymbol{y}}_i-\hat{\boldsymbol{e}}_i}{1+\lambda}. \tag{12}$$

(2) Fixing $\boldsymbol{z}$ and computing $\boldsymbol{e}$

The problem is

$$\frac{1}{2}\left\|\left(\overline{\boldsymbol{y}}_i-\boldsymbol{U}\hat{\boldsymbol{z}}_i\right)-\boldsymbol{e}_i\right\|_F^2+\beta\left\|\boldsymbol{e}_i\right\|_1. \tag{13}$$

Then the optimal solution $\hat{\boldsymbol{e}}_i$ can be computed by

$$\hat{\boldsymbol{e}}_i=\boldsymbol{S}_\beta\left(\overline{\boldsymbol{y}}_i-\boldsymbol{U}\hat{\boldsymbol{z}}_i\right), \tag{14}$$

where $\boldsymbol{S}_\beta(\boldsymbol{x})$ is the soft-thresholding operator and

$$\boldsymbol{S}_\beta\left(\boldsymbol{x}\right)=\max\left(\left|\boldsymbol{x}\right|-\beta,0\right)*\mathrm{sgn}\left(\boldsymbol{x}\right). \tag{15}$$

By iteratively executing the steps (1) and (2), the optimization can be solved efficiently. And the progress will keep running until the stop criterion is met (like reaching the maximum of iteration or the difference of objective function values less than some threshold). Tab.1 is the summary of the iterative method.

Tab.1 Summary of the iterative method

Input: observation vector $\overline{\boldsymbol{y}}_i$, base set $\boldsymbol{U}$, constant λ and constant β
Initialize: j=1, $\boldsymbol{e}_i^{(j)}=0$
Iterate:
Obtain $\boldsymbol{z}_i^{(j+1)}$ via $\boldsymbol{z}_i^{(j+1)}=\boldsymbol{U}^{\mathrm{T}}\left(\overline{\boldsymbol{y}}_i-\boldsymbol{e}_i^{(j)}\right)/\left(1+\lambda\right)$
Obtain $\boldsymbol{e}_i^{(j+1)}$ via $\boldsymbol{e}_i^{(j+1)}=\boldsymbol{S}_\beta\left(\overline{\boldsymbol{y}}_i-\boldsymbol{U}\boldsymbol{z}_i^{(j+1)}\right)$
$j\leftarrow j+1$
Until reaching the stop criterion
Output $\hat{\boldsymbol{z}}_i$, $\hat{\boldsymbol{e}}_i$

It is clear that

$$L\left(\boldsymbol{z}_i^{(j+1)},\boldsymbol{e}_i^{(j+1)}\right)<L\left(\boldsymbol{z}_i^{(j+1)},\boldsymbol{e}_i^{(j)}\right)<L\left(\boldsymbol{z}_i^{(j)},\boldsymbol{e}_i^{(j)}\right). \tag{16}$$

Thus, the iterative method can approximate a local minimal value. What's more, as the optimization model is convex, the iterative method can obtain the global minimal solution.

After computing the coefficient $\boldsymbol{z}$ and residual $\boldsymbol{e}$ for candidates, we evaluate each candidate by

$$\boldsymbol{S}_i=\frac{1}{2}\left\|\left(\overline{\boldsymbol{y}}_i-\boldsymbol{U}\hat{\boldsymbol{z}}_i\right)-\hat{\boldsymbol{e}}_i\right\|_F^2+\beta\left\|\hat{\boldsymbol{e}}_i\right\|_1,$$

$$i=1,2,...,n. \tag{17}$$

Furthermore, the observation model of each candidate is

$$p\left(\boldsymbol{y}_{i,t}\left|\boldsymbol{x}_{i,t}\right.\right)=\exp\left(-S_i\right). \tag{18}$$

Then we select the $\hat{i}$-th candidate as the tracking result at frame t, which follows

$$\hat{i}=\arg\max_i p\left(\boldsymbol{y}_{i,t}\left|\boldsymbol{x}_{i,t}\right.\right). \tag{19}$$

After the tracking result is obtained by frames, we construct a set and collect the results into the set. When we collect enough results, we employ the incremental PCA learning method[5] to compute the newer PCA subspace base set $\tilde{\boldsymbol{U}}$ and centre $\bar{\boldsymbol{\mu}}$. Assume that at frame t, the $\hat{i}$-th candidate is selected as the target, and the corresponding residual matrix is $\boldsymbol{e}_{\hat{i}}$, the image patch is $\boldsymbol{y}_{\hat{i}}$. In order to construct the set of the results, we compute the non-zero ratio η of the residual $\boldsymbol{e}_{\hat{i}}$, and then compare η with two predefined thresholds, lower threshold *low* and higher threshold *high*. We collect the results in three different ways, such as fully, partly and no update cases.

(1) η<*low*, which means the target is not occluded, the result $\boldsymbol{y}_{\hat{i}}$ doesn't contain much noise and the result can be well represented by the base set $\boldsymbol{U}$ and centre $\boldsymbol{\mu}$, then we can directly collect $\boldsymbol{y}_{\hat{i}}$ into the set.

(2) *low*<η<*high*, which means the result contains noise. The reason is that the target may change obviously or be partially occluded. If we directly collect the result, noise will be included into the base, which will cause tracking drift. Consequently, we construct a new sample $\tilde{\boldsymbol{y}}_t$ in the following way:

$$\tilde{\boldsymbol{y}}_t(q) = \begin{cases} \boldsymbol{\mu}(q) & \text{if } \boldsymbol{e}_{\hat{i}}(q) \neq 0 \\ \boldsymbol{y}_{\hat{i}}(q) & \text{if } \boldsymbol{e}_{\hat{i}}(q) = 0 \end{cases} \quad q = 1,2,...,d \ . \tag{20}$$

And then we collect $\tilde{\boldsymbol{y}}_t$ as the result of frame t.

(3) η>*high*, which means the target suffers heavy change or occlusion and the result contains much noise. Therefore, we do not collect the result at frame t.

After collecting enough results, we adopt the incremental PCA learning method to learn the new base set $\tilde{\boldsymbol{U}}$ and centre $\bar{\boldsymbol{\mu}}$.

The proposed algorithm is implemented on the platform as follows: CPU is i3-3220 with 3.3 GHz, system is Win7, and Matlab version is v2013. To evaluate the performance, we test the proposed algorithm on 9 challenging sequences available in Ref.[18]. And we also compare the results with other 5 state-of-art methods, including IVT[5], L1APG[14], SCM[19], TLD[20], MTT[21]. The source codes of the 5 methods are all provided by the authors. For a fair comparison, in each test sequence, the location of the target is labelled manually at the first frame and it is the same for all methods. The parameters of our algorithm are as follows: the number of the candidates n is 600, the number of the bases m is 11, image patch is resized to 32×32, d is 1 024, λ is 0.25, β is 0.3, *low* is 0.1 and *high* is 0.6. We update the base and mean template every 5 frames. To quantify the performance of methods, we employ the average centre location error (*ACLE*) and the average overlap rate (*AOR*)[22] as the criteria. For a robust method, the value of *ACLE* should be close to 0 and the *AOR* should be close to 1. Fig.1 shows partial tracking results of all the methods on the 9 sequences. Tab.2 shows the *ACLE* results of 6 methods on 9 sequences. Tab.3 shows the *AOR* results. Tab.4 shows the efficiency of each method, and bigger value of the frames per second (fps) means that the method is more efficient.

—IVT—L1APG—TLD—SCM—MTT—Ours

Fig.1 Partial tracking results of 6 methods on 9 sequences of BlurCar2, BlurFace, Car4, CarDark, Dudek, FaceOcc1, Jumping, Singer1 and Walking, respectively

Tab.2 *ACLE* (pixel) of the methods

Sequence	IVT	L1APG	TLD	SCM	MTT	Ours
BlurCar2	155	144	6.5	125	141	3.5
BlurFace	157	17	3.7	108	80	7.6
Car4	2.1	77	12	4.2	22.3	1.7
CarDark	8.4	1.0	27	1.2	1.6	1.2
Dudek	9.6	23.5	18.1	10.7	14.0	8.5
FaceOcc1	18.4	17.3	27.3	13.3	20.9	13.0
Singer1	11.3	53	8.0	2.7	36.1	3.3
Jumping	61	83	5.9	68	84.5	4.7
Walking	1.6	3.3	10	2.5	3.5	1.8
Average	47.16	46.57	13.17	37.29	44.88	5.03

Tab.3 *AOR*(%) of the methods

Seuence	IVT	L1APG	TLD	SCM	MTT	Ours
BlurCar2	14.0	12.1	73.0	18.9	11.8	88.9
BlurFace	13.7	73.4	88.0	18.4	33.7	82.5
Car4	87.5	24.8	63.2	75.7	44.6	90.2
CarDark	66.3	88.4	44.8	84.3	82.6	86.2
Dudek	75.2	69.1	64.7	76.8	75.8	71.7
FaceOcc1	72.6	75.2	58.4	79.3	70.1	79.7
Singer1	57.3	28.4	72.5	86.8	34.0	75.5
Jumping	12.2	15.0	66.3	12.2	19.6	63.4
Walking	76.6	75.2	44.6	71.1	66.5	74.3
Average	52.82	51.29	63.94	58.17	48.74	79.16

Tab.4 Frames per second (fps) of the methods

Method	IVT	L1APG	TLD	SCM	MTT	Ours
fps	19.4	0.8	12.1	0.3	0.25	6.5

In the sequences of BlurCar2, BlurFace and Jumping, there are fast motion and motion blur. When the target has fast motion and obvious scale variation simultaneously, the information of the target is reduced sharply. TLD learns the variation with some noise. IVT, L1APG, SCM and MTT cannot wrap enough information and they match the background as the target. The proposed algorithm is based on the particle filter and uses the residual $\boldsymbol{e}$ to evaluate the candidates. Consequently, the proposed algorithm can select the target accurately.

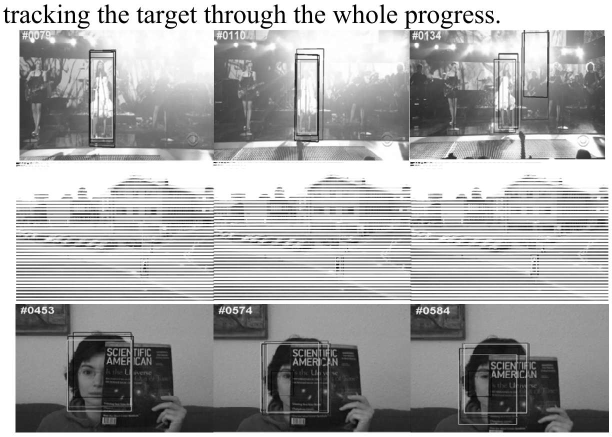
In the sequence of Car4, there are illumination and scale variation. SCM and TLD lose the target when the target suffers severe illumination variation. L1APG and MTT cannot decide whether the variation is noise and they locate the wrong position as the target. IVT and the proposed algorithm can track the target in the whole tracking progress. Because they are both based on the PCA subspace, and they can well handle the illumination variation.

In the sequence of CarDark, there is mainly background clutter. TLD and IVT fail because the target is similar to the background, and there is limited information for distinguishing. Other methods are all based on the sparse representation and they can accurately represent the target and the background respectively.

In the sequence of Dudek, there are out-of-plane rotation and scale variation. The proposed algorithm and TLD can describe the target accurately. Other methods have error in description of the scale. Because when the target is out-of-sight, L1APG, SCM, MTT and IVT directly update the template with the error. On the contrary, the proposed algorithm makes use of the residual $\boldsymbol{e}$ to detect the state of the target and adopts a flexible update scheme. In this way, the proposed algorithm can obtain a better result.

In the sequences of Singer1, FaceOcc1 and Walking, there is occlusion. In the Singer1 sequence, the lightness of the stage in the background gradually increases and covers the body of the singer. In the Walking sequence, the target is occluded partly by a pillar in a short time. In the FaceOcc1 sequence, the human face is severely occluded by the book. From the results, we can see that IVT and TLD are sensitive to the occlusion. L1APG and MTT are more robust than the IVT. L1APG and MTT are based on the sparse representation, so they are robust to the occlusion but sensitive to the illumination. The proposed algorithm and SCM track the target well in all three sequences. The proposed algorithm integrates the advantages of the sparse representation and the subspace representation. It can both adapt to the illumination variation and occlusion. From Fig.2, we can see that the proposed algorithm is more robust than L1APG and IVT when dealing with the situation of illumination variation and occlusion.

Compared with the sparse representation based methods, like L1APG, MTT and SCM, the proposed algorithm takes the advantage of the subspace representation, which makes the algorithm more robust to illumination variation. Compared with the subspace representation based methods (e.g. IVT), the proposed algorithm takes the advantage of the sparse representation, which makes the algorithm more robust to occlusion and background clutter. Furthermore, the proposed algorithm updates the dictionary in different ways according to the state of the target and the non-zero ratio of the residual $\boldsymbol{e}$, which makes the proposed algorithm adapt to the variation of the target and keep tracking the target through the whole progress.



(a) (b) (c)

—IVT—L1APG—TLD—SCM—MTT—Ours

Fig.2 Partial results of three sequences: (a) Singer1; (b) Walking; (c) FaceOcc1

We test the proposed algorithm in all the sequences with different predefined thresholds. We use the results of the Car4 and Singer1 sequences as the example to illustrate the performance. Tab.5 shows partial results of the Car4 sequence and Tab.6 shows partial results of the Singer1 sequence. In the tables, results of some thresholds have the same value, so we omit them for brevity, likely the values of 0.6, 0.7, 0.8 and 0.9 of the *low* threshold in Tab.5 and others. From the results shown in Tab.5, we can see that the proposed algorithm obtains the best result when *low* is 0.1 and *high* is 0.3. But in Tab.6, the proposed algorithm obtains the best result when *low* is 0.2 and *high* is 0.4. In different sequences, the proposed algorithm obtains the best result in different thresholds. Consequently, in order to handle with more tracking situations, we can set *low* as 0.1 and *high* as 0.7.

Tab.5 Partial results of different predefined thresholds in Car4 sequence

low / *high*	0.1	0.2	0.3	0.4	0.5	1.0
0.1	152.7	148.1	150.9	1.9	1.9	1.9
0.2	1.9	148.1	150.9	1.9	1.9	1.9
0.3	1.7	1.9	150.9	1.9	1.9	1.9
0.4	1.7	151.2	152.2	1.9	1.9	1.9
0.5	1.7	150.8	153.1	1.9	1.9	1.9
0.6	1.7	138.1	153.2	1.9	1.9	1.9
0.7	1.7	137.2	151.5	1.9	1.9	1.9
1.0	1.7	140.2	152.6	1.9	1.9	1.9

Tab.6 Partial results of different predefined thresholds in Singer1 sequence

high \ low	0.1	0.2	0.3	0.4	0.5	1.0
0.1	140.1	129.2	3.9	3.9	8.6	8.6
0.2	140.5	129.2	3.9	3.9	8.6	8.6
0.3	118.9	130.2	3.9	3.9	8.6	8.6
0.4	2.8	2.7	3.9	3.9	8.6	8.6
0.5	3.3	3.2	8.8	8.6	8.6	8.6
0.6	4.3	2.9	8.8	8.6	8.6	8.6
0.7	3.3	2.9	8.8	8.6	8.6	8.6
1.0	3.3	2.9	8.8	8.6	8.6	8.6

In this letter, we propose an algorithm which integrates the advantages of the subspace representation and the sparse representation. The algorithm is implemented in the particle filter framework. According to the results of comparison, we can conclude that the proposed algorithm is more robust than the other methods, and it can handle with some severe situations encountered during the tracking progress.

References

[1] Wang D, Yang J, Yang L and Ge H, Journal of Optoelectronics·Laser **27**, 1066 (2016). (in Chinese)

[2] Li C, Shi J, Xu L and Wei L, Journal of Optoelectronics·Laser **26**, 1381 (2015). (in Chinese)

[3] Yang H, Wu X, He B and Zhu M, Journal of Optoelectronics·Laser **26**, 170 (2015). (in Chinese)

[4] Bai T. and Li Y., IEEE Trans. on Industrial Electronics **10**, 538 (2014).

[5] Ross D. A., Lim J., Lin R. S. and Yang M. H., International Journal of Computer Vision **77**, 125 (2008).

[6] Turk M. A. and Pentland A. P., J. Cogn. Neurosci. **3**, 71 (2001).

[7] Shirazi S., Harandi M., Lovell B.C. and Sanderson C., Object Tracking via Non-euclidean Geometry: A Grassmann Approach, Proc. IEEE Winter Conf. on Applications of Comput. Vision (WACV), 901 (2014).

[8] Kumar B.S., Swamy M.N.S. and Ahmad M.O., Weighted Residual Minimization in PCA Subspace for Visual Tracking, IEEE International Symposium on Circuits and Systems, 986 (2016).

[9] Wright J., Yang A., Ganesh A., Sastry S. and Ma Y., IEEE Transactions on Pattern Analysis & Machine Intelligence **31**, 210 (2008).

[10] Yang J., Wright J., Huang T. S. and Ma Y., IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society **19**, 2861 (2010).

[11] Cevher V., Sankaranarayanan A., Duarte M. F., Reddy D., Baraniuk R. G. and Chellappa R., Compressive Sensing for Background Subtraction, European Conference on Computer Vision, Springer Berlin Heidelberg, 155 (2008).

[12] Mei X., Ling H. B., Wu Y., Blasch E. and Bai L., Minimum Error Bounded Efficient L1 Tracker with Occlusion Detection, The 24th IEEE Conference on Computer Vision and Pattern Recognition, 1257 (2011).

[13] Mei X. and Ling H. B., Robust Visual Tracking using l1 Minimization, IEEE International Conference on Computer Vision, 1436 (2009).

[14] Bao C. L., Wu Y., Ling H. B. and Ji H., Real Time Robust L1 Tracker using Accelerated Proximal Gradient Approach, IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 1830 (2012).

[15] Zhuang B.H., Lu H.C., Xiao Z.Y. and Wang D., IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society **23**, 1872 (2014).

[16] Wang D., Lu H.C. and Yang M.H., Least Soft-threshold Squares Tracking, IEEE Computer Vision and Pattern Recognition, 2371 (2013).

[17] Xiao Z.Y., Lu H.C. and Wang D., IEEE Transactions on Circuits & Systems for Video Technology **24**, 1301 (2014).

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

[19] Zhong W., Lu H.C. and Yang M.H., Robust Object Tracking via Sparsity-based Collaborative Model, IEEE Computer Vision and Pattern Recognition, 1838 (2012).

[20] Kalal Z., Mikolajczyk K. and Matas J.R., IEEE Transactions on Pattern Analysis & Machine Intelligence **34**, 1409 (2012).

[21] Zhang T., Ghanem B. and Liu S., International Journal of Computer Vision **101**, 367 (2013).

[22] Everingham M., Van Gool L., Williams C.K., Winn J. and Zisserman A., International Journal of Computer Vision **88**, 303 (2010).