Signal Processing 110 (2015) 178–187

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Single object tracking via robust combination of particle filter and sparse representation

Shuangyan Yi [a], Zhenyu He [a,*], Xinge You [b], Yiu-Ming Cheung [c,d]

[a] School of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, China
[b] Department of Electronics and Information Engineering, Huazhong University of Science and Technology, China
[c] Department of Computer Science, Hong Kong Baptist University, Hong Kong
[d] United International College, Beijing Normal University, Hong Kong Baptist University, Zhuhai, China

ARTICLE INFO

Article history:
Received 12 May 2014
Received in revised form
13 September 2014
Accepted 17 September 2014
Available online 18 October 2014

Keywords:
Visual object tracking
Sparse representation
Occlusion prediction
Template update
Particle filter

ABSTRACT

The drifting problem is a core problem in single object tracking and attracts many researchers' attention. Unfortunately, traditional methods cannot well solve the drifting problem. In this paper, we propose a tracking method based on the robust combination of particle filter and reverse sparse representation (RC-PFRSR) to reduce the drifting. First, we find the ill-organized coefficients. Second, we propose a diagonal matrix $\boldsymbol{\alpha}$, whose diagonal line includes each patch contribution factor, to function each patch coefficient value of one candidate obtained by sparse representation. Third, we adaptively discriminate the power of each patch within the current candidate region by an occlusion prediction scheme. Our experimental results on nine challenging video sequences show that our RC-PFRSR method is effective and outperforms six state-of-the-art methods for single object tracking.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Single object tracking, which plays an indispensable role in motion analysis, activity recognition, video surveillance and traffic monitoring, is a basic work in vision community and has achieved many progresses recently. While, it is still a challenging task to design a robust visual tracking method because of many negative factors existing in video sequences, such as occlusions, background variations and pose variations. Among these factors, the drifting is one core problem and has not been solved thoroughly.

Before the popularity of sparse model, the particle filter (PF), which is a classical framework in the field of single object tracking, is paid many attentions. Usually, approaches for single object tracking can be divided into two categories: discriminative models and generative models. The discriminative models [1–4] aim to discriminate the target from the background by training a classifier according to the information from both the target and the background. While, the generative models [5–8] aim to search for regions, which are extremely similar with the true object targets, based on templates or subspace. In recent years, sparse representation, which origins from compression sensor [9,10], has been successfully applied on face recognition (FR) [11] and has highly attracted researchers' interests on object tracking because of its ability on feature representation. And then, the sparse representation, which is used to reconstruct the object target by matching templates with the minimum reconstruction error, starts to play an extremely essential role in single object tracking. Generally, single object tracking approaches, based on the sparse representation, can be divided into three categories: the sparsity-based discriminative classifier (SDC), the sparsity-based generative model

* Corresponding author.
E-mail addresses: zyhe@hitsz.edu.cn (Z. He), youxg@hust.edu.cn (X. You), ymc@comp.hkbu.edu.hk (Y.-M. Cheung).

http://dx.doi.org/10.1016/j.sigpro.2014.09.020
0165-1684/© 2014 Elsevier B.V. All rights reserved.

(SGM) [12–14], and the combination of SGM and SDC [15,16]. For example, Mei and Ling first applied sparse representation for tracking in [12], in which, the object target candidate (it corresponds to one particle and means the probable location of the object target in next frame) is represented as a linear combination of target templates and trivial templates, and every trivial template has only one nonzero element. Moreover, the good object target candidate is assumed to be sparsely represented by the learned template set and finally this sparse optimization problem is solved as a L1 minimization problem with no negative constraints. Mei et al. [13] proposed a bounded particle resampling (BPR)-L1 tracker based on sparse representation by combining PF framework to improve the tracker speed. Liu et al. [15] developed a robust tracker based on sparse representation by combining generative model with discriminative model. However, the above methods have a detrimental effect on the quality of tracking when the video sequences face with the partial occlusion and background clutter. With respect to this problem, Xu et al. [14] proposed a method based on adaptive structural local sparse appearance (ASLSA) and obtained a relatively optimal tracking result. Moreover, Zhong et al. [16] first proposed the SDC and then combined it with SGM to get the combination of SGM and SDC.

The tracking methods based on sparse representation [12–16] can be summarized as

$$\overline{\mathbf{b}} = \min_{\mathbf{b}} \|\mathbf{Y}-\mathbf{D}\mathbf{b}\|_2^2 + \lambda\|\mathbf{b}\|_p, \tag{1}$$

where $\|\mathbf{Y}-\mathbf{D}\mathbf{b}\|_2^2$ means the error term, $\mathbf{Y}\in\mathbb{R}^d$, $\mathbf{D}$ is dictionary, $\mathbf{b}$ is the coefficient corresponding to $\mathbf{Y}$ and $\|\mathbf{b}\|_p$ means the sparse term. The tradeoff between sparse term and reconstruction error term is governed by the parameter λ. In detail, when $p=0$, Eq. (1) becomes a NP-hard problem; when $p=2$, the sparsity of the sparse term in Eq. (1) becomes weak; when $p=2,1$, Eq. (1) has the property of row sparsity; when $p=1$, Eq. (1) is actually a LASSO problem [17] which gets its popularity as a convex optimization problem and has been accepted as a most useful tool in different fields [18–24]. Furthermore, it also can be applied into object tracking by solving Eq. (2) [12]:

$$\begin{aligned}\overline{\mathbf{b}} &= \min_{\mathbf{b}} \quad \|\mathbf{Y}-\mathbf{D}b\|_2^2 + \lambda\|\mathbf{b}\|_1,\\ &\text{s.t.} \quad \mathbf{b}\geqslant\mathbf{0}\end{aligned} \tag{2}$$

where $\mathbf{Y}\in\mathbb{R}^d$ (d is the dimension of the image vector) means one candidate, $\mathbf{D}\in\mathbb{R}^{d\times n}$ means the dictionary composed of tracking results from n frames, λ, which usually satisfies $\lambda>0$, controls the sparsity of the solution [25] and $\mathbf{b}\geqslant\mathbf{0}$ means all the elements in $\mathbf{b}$ are nonnegative. Second, the useful coefficients, which cannot represent the object target invalidly, are abstracted from the trivial coefficients [26]. Third, target templates are replaced by PCA basis [27] in order to avoid the redundancy of templates set and can represent the object target effectively. Fourth, the patch-dictionary, which is formed by dividing each template into N overlapped local image patches [16,14], is paid more and more attentions by researchers. Moreover, the more description about patch term can refer to Section 2.

Traditional tracking methods usually use Eq. (2) to reconstruct the candidate by matching dictionary composed of n tracking templates. While, we use reverse sparse representation, which is shown in Eq. (3), to reconstruct the template set by matching dictionary composed of candidates.

To the best of our knowledge, the ASLSA method [14], compared with other state-of-the-art methods, has obtained better partial information by assembling the patches at the same positions of the patch-dictionary. Unfortunately, the ASLSA method still suffers from the drifting because it does not consider geometric information between patches and assigns the same contribution factor for every patch within one candidate region. Generally speaking, the drifting starts to appear when the right candidate is not suitably represented instead of the false negative candidate. Therefore, the drifting is very easy to occur under an occlusion case. In addition, the drifting also tends to appear if the way of dividing patches is not suitable. Hence, the core scale of dividing patches is that the patches should be more discriminative. Therefore, we adopt our own patch way for some datasets, which is described in the experimental section, to focus on the key foreground appearance information.

Therefore, in this paper, we propose a robust combination of particle filter and reverse sparse representation (RC-PFRSR) method to reduce the drifting. In summary, our key contribution in this paper is to explore how to integrate the partial information into the integral one to improve the tracking result. That is, our method exploits the spatial information of each local patch with an occlusion prediction scheme. Different with the above proposed methods, our RC-PFRSR method, the framework of which is shown in Fig. 1, first adds a contribution factor $\boldsymbol{\alpha}$ (translation matrix), which is inspired by investigating the origin coefficients with occlusion information, into the object function in [14] to reduce the drifting. In mathematics, the function of $\boldsymbol{\alpha}$ is to make the dictionary have a parallel translation, thereby represent the candidate in the best. In this way, the value of object function approximates zero and the reconstruction error reaches

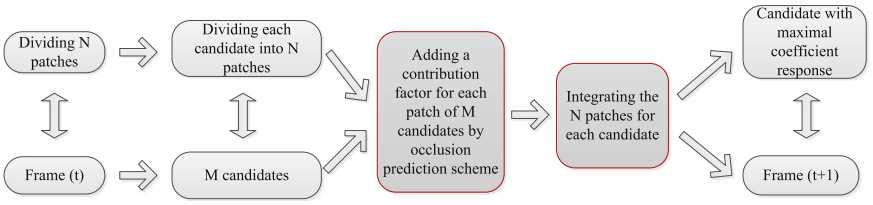

Fig. 1. The framework of our tracking method. The part boxed by a red line is our core idea. We use Frame (t) and Frame ($t+1$) to represent the tracking result in frame t and $t+1$, respectively. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

the minimum. Second, we can adaptively discriminate the power of patches within the current candidate region by an occlusion prediction scheme.

The remainder of this paper is organized as follows. In Section 2, we use all candidates of each frame to reconstruct templates by reverse sparse representation (RSR). In Section 3, we propose our RC-PFRSR method to reduce the drifting. In Section 4, we make quantitative and qualitative evaluations for the RC-PFRSR method, and compare it with several art-of-the-state methods. Finally, the conclusion is drawn in Section 5.

2. Reverse Sparse Representation (RSR)

First, the tracking results from the previous n frames are selected as a template-dictionary. This template-dictionary is represented by $\mathbf{T}=\mathbf{T}_1 \cup \mathbf{T}_2 \cup \ldots \cup \mathbf{T}_n$, where $\mathbf{T}_i \in \{1,2,\ldots,n\}$ represents the tracking result of each frame and n is the number of templates. Then each of the tracking result $\mathbf{T}_i$ is divided into N patches, where N is the number of local patches within the object target region. In this way, a patch template is obtained, which is denoted as $\mathbf{D}=[\mathbf{D}_1^1, \mathbf{D}_2^1, \ldots, \mathbf{D}_{N(i-1)+j}^i, \ldots, \mathbf{D}_{(n\times N)}^n] \in \mathbb{R}^{d\times(n\times N)}$

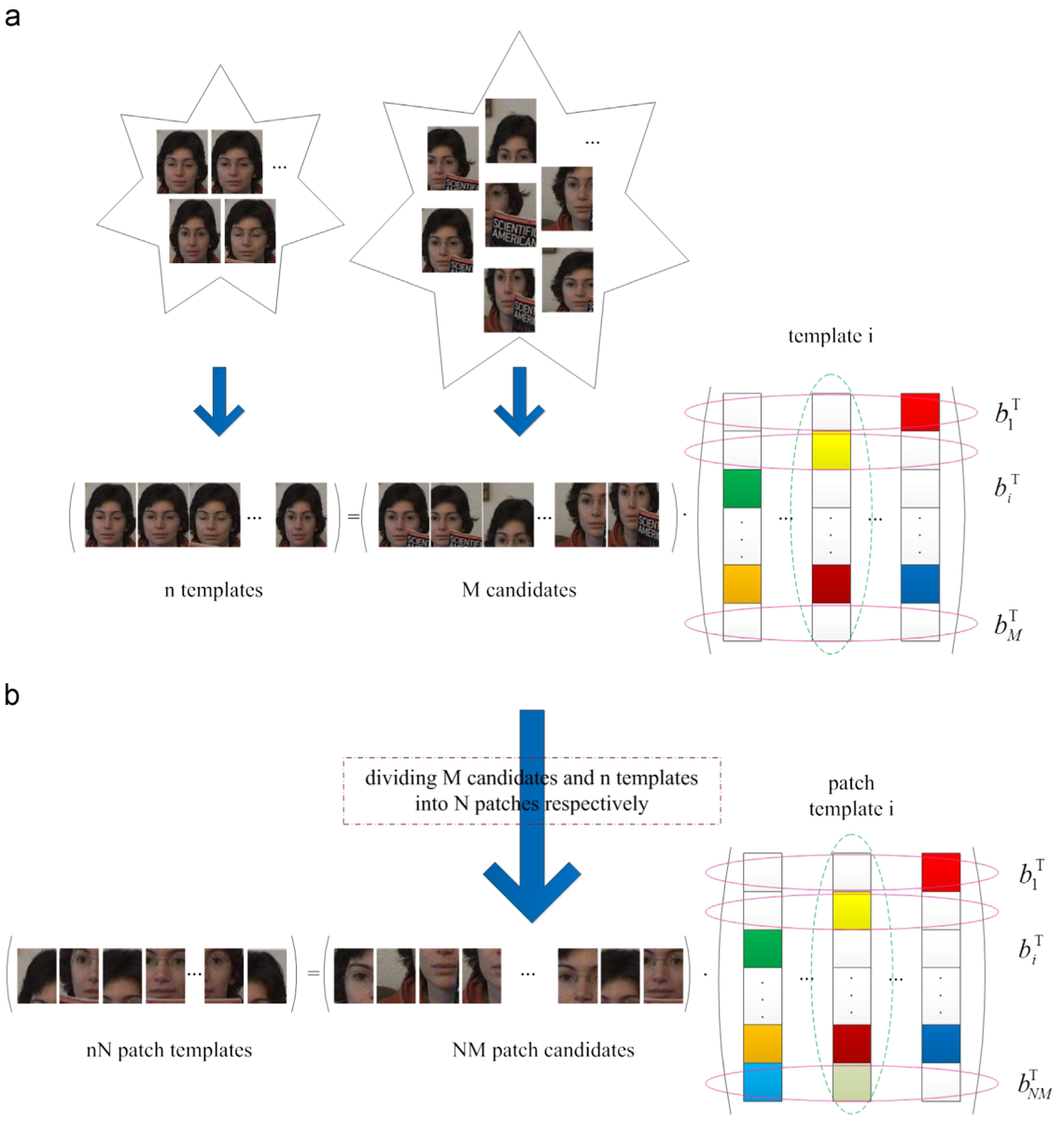

Fig. 2. One example to show our method. These pictures come from *Faceocc*1 sequence (a) means that M candidates are reconstructed by n templates and then one optimal candidate is chose which has the most little reconstruction error while (b) are the formulation of patches correspondingly. The rightmost part of this figure (b) is denoted as matrix **b**, and each column vector of it, which is shown as the green virtual oral line, means the sparse coefficients of all NM patch candidates (M candidates) representing one patch template i (one template i) while the pink oral line means the responses of one patch candidate (one candidate) representing on all patch templates (templates). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

according to the increasing patch location and template number. Here, d means the dimension of the image patch vector (i.e., the size of each patch), $i \in \{1,2,\ldots,n\}$ means the number of templates, and $j \in \{1,2,\ldots,N\}$ means the j-th location number of one patch of one template. In this paper, we adopt M candidates, which surround with the current object target, and use $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_M,\ldots,\mathbf{y}_{NM}]$, which is formed by dividing N patches for each candidate and $\mathbf{y}_i \in \mathbb{R}^d$, to represent the patch-dictionary. Different with the traditional object function such as [14,12], we use M candidates to reconstruct the templates in each frame. Therefore, our object function is shown reversely as follows for each frame.

$$\overline{\mathbf{b}} = \min_{\mathbf{b}} \quad \|\mathbf{D}-\mathbf{Y}b\|_2^2+\lambda\|\mathbf{b}\|_1, \quad \text{s.t.} \quad \mathbf{b}\succcurlyeq\mathbf{0} \tag{3}$$

where the matrix $\mathbf{b} \in \mathbb{R}^{NM\times(n\times N)}$, got by Eq. (3), is the corresponding sparse coefficients of M candidates assembled with NM local patches and is shown in the rightmost part of Fig. 2(b). And $\mathbf{b}\succcurlyeq 0$ means each $\mathbf{b}_i$ is nonnegative. Note that each row of $\mathbf{b}$ can be denoted as $\mathbf{b}_i^\top=[\mathbf{b}_i^{(1)\top},\mathbf{b}_i^{(2)\top},\ldots,\mathbf{b}_i^{(n)\top}] \in \mathbb{R}^{1\times(n\times N)}$, $i \in \{1,2,\ldots,NM\}$, where $\mathbf{b}_i^{(k)\top}=\{b_i^{(k1)},b_i^{(k2)},\ldots,b_i^{(kN)}\} \in \mathbb{R}^{1\times N}$ denotes the k-th group of the coefficient vector $\mathbf{b}_i$. After the sparse coefficients of every local patch are divided into several groups according to the same one patch position across n templates, these grouped coefficients $\mathbf{v}_i$ for the i-th patch can be computed as follows:

$$\mathbf{v}_i=\frac{1}{C}\sum_{k=1}^{n}\mathbf{b}_i^{(k)}, \quad i=1,2,\ldots,N, \tag{4}$$

where the vector $\mathbf{v}_i$ corresponds to the i-th local patch and C is a constant. As the templates contain the object target with some appearance variations, the patch blocks that appear frequently in these templates should be weighted more than the others, thereby get the more robust representation by Eq. (4). Therefore, the local appearance variation can be clearly described by the patch blocks at the same positions of the patch-dictionary.

At this moment, a square matrix $\mathbf{V}$, denoted by $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_N]$, is formed by all the local patch vector $\mathbf{v}_i$ within a candidate region as the pooled features, and then is further processed with an alignment-pooling way. Finally, we get the most optimal candidate $\overline{\mathbf{b}}$ from candidate set $\mathbf{S}=\{\mathbf{S}_1,\mathbf{S}_2,\ldots,\mathbf{S}_i,\ldots,\mathbf{S}_M\}$, which contains M candidates. It deserves to note that the most optimal candidate might not be the right candidate because of the limitation of this method.

We should point out the difference between Eqs. (2) and (3) is that one is to obtain the most optimal candidate by using the n templates to reconstruct M candidates while the other is to obtain the most optimal candidate by using the M candidates to reconstruct n templates.

3. Robust Combination of Particle Filter and Reverse Sparse Representation (RC-PFRSR)

In this section, we first describe the tracking framework (see Algorithm 1) by designing a diagonal matrix $\boldsymbol{\alpha}$. Then, we describe the algorithms used for occlusion patch location detection (see Algorithm 2) and template update (see Algorithm 3), which are two important parts of the tracking framework.

3.1. Tracking framework

As usual, object tracking is carried out in the classical PF framework. Specifically, x_t is denoted as the state variable representing the location and shape of an object target in frame t. And the goal of the tracking problem is to estimate the state probability $p(x_t|z_{1:t})$, where $z_{1:t}=\{z_1,\ldots,z_t\}$ represents the observation set of the object targets up to t frames. Usually, a Bayesian sequential estimation is used to describe the tracking proceeds as follows:

$$p(x_t|z_{1:t}) \propto p(z_t|x_t)\int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})\,dx_{t-1} \tag{5}$$

In this paper, we use x_t^j to represent the j-th candidate of state status x_t in frame t, $j \in \{1,2,\ldots,M\}$, and our goal becomes to estimate the following formula:

$$\overline{x}_t=\max_{x_t^j} p(z_t^j|x_t^j)p(x_t^j|x_{t-1}) \tag{6}$$

where z_t^j means the observation vector and has a one-to-one corresponding to the observation vector of x_t^j. And $p(x_t^j|x_{t-1})$ means a dynamic model in the successive frames, which is applied to describe the temporal correlation of the object target states and is assumed to be constrained with a Gaussian distribution $\boldsymbol{N}(x_t^j|x_{t-1},\sigma)$. Therefore, M candidates are generated according to the dynamic model $p(x_t^j|x_{t-1})$. Note that the affine transformation with six parameters $x_t=(x,y,\theta,s,\beta,\phi)$ is used to model the object target motion between these two consecutive frames, in which, x,y denote the coordinates, θ,s,β,ϕ denote rotation angle, scale, aspect ratio and skew respectively. While $p(z_t^j|x_t^j)$ means an appearance model in the fixed t frame for candidate $\mathbf{S}^j$ and represents the likelihood state of z_t^j generated from state x_t^j. Therefore, the appearance model is casted as the most crucial component, which is based on the dynamic model. Hence, our goal becomes to maximize $p(z_t^j|x_t^j)$. Before we express $p(z_t^j|x_t^j)$, we first design the contribution matrix $\boldsymbol{\alpha}$ for all the candidates of each frame, of which each element in diagonal line α_k, $k \in 1,2,\ldots,N$ means the contribution factor of each corresponding patch within a candidate.

$$\mathbf{f}^j=\boldsymbol{\beta}^j \odot \mathbf{v}^j \tag{7}$$

where $\odot$ is the Hadamard product (element-wise product). And the vector $\boldsymbol{\beta}^j=diag(\boldsymbol{\alpha})$, $\mathbf{v}^j=diag(\mathbf{V}^j)$, whose elements come from the diagonal line of a diagonal matrix, mean the contribution value and coefficient value of the corresponding patch for candidate $\mathbf{S}_j$. $\mathbf{V}^j$ means a square matrix which is formed by all the vectors $\mathbf{v}_k^j$ of local patches in a candidate region, which is computed as Eq. (4). Therefore, $\mathbf{f}^j$ is regarded as the pooled feature by taking the diagonal elements of the square matrix $\boldsymbol{\alpha}^j\mathbf{V}^j$, where $\mathbf{f}^j=\{f_1^j,f_2^j,\ldots,f_k^j,\ldots,f_N^j\}$.

Therefore, $p(z_t^j|x_t^j)$ is alternatively approximated by a set of N patches $\{v_{kk}^j\}_{k=1}^N$ with the corresponding contribution

factor α_k^j as follows:

$$\max_j p(z_t^j|x_t^j) \propto \max_j \sum_{k=1}^{N} f_k^j = \max_j \left\{ \sum_{k=1}^{N} \alpha_k^j v_{kk}^j \right\}$$
$$= \max \left[f_{\mathrm{rsp}}(\boldsymbol{\alpha}, 1, N^2) \right] f_{\mathrm{E}} \left[f_{\mathrm{rsp}}(\rho \mathbf{b}, N^2, 1) \right], \quad (8)$$

where α_k^j means a scalar weight for the j-th candidate $\mathbf{S}_j$, and v_{kk}^j, which comes from the k-th element of vector $\mathbf{v}_k^j$ computed as Eq. (4), means the partial information of patch k from candidate $\mathbf{S}_j$. In practice, the right side of the equation denotes the similarity between the most optimal candidate and the right object target.

What's more important, the more details of tracking based on the tracking framework is shown in Algorithm 1. The final tracking results are related to the several parameters such as λ, n, and M in Algorithm 1. In detail, $\lambda = 0.01$ is the empirically setting value, n is the number of templates in template set, and M is the number of candidates in each frame. For n or M, the value is the larger, the tracking results are more accurate but can incur the high computational complexity. Therefore, in our experiments, we set $\lambda = 0.01$, $n = 15$, and $M = 600$, respectively and empirically.

Algorithm 1. The tracking framework.

Require:

Set $\lambda = 0.01$, $n = 15$, $M = 600$, and take center point *param*0 as the object target in first frame; Get the tracking results in these n frames by using KDTree algorithm; Abstract these tracking results as template set $\mathbf{T}$ and then obtain patch-template $\mathbf{D}$ by dividing patches for them

Tracking result for $t = n+1$

1: Take M candidates surrounding with the tracking result in n-th frame and crop them into patches to formulate the patch-dictionary $\mathbf{Y}$ and then solve Lasso problem Eq. (3)
2: Associate with $\boldsymbol{\alpha} = \mathbf{I}_{N \times N}$ at frame $n+1$ and then adopt the alignment-pooling algorithm to ascertain the most optimal candidate

Tracking results from $t = n+2$ **to the end of the sequence**

3: Take M candidates surrounding with the tracking result in $(t-1)$-th frame to formulate the patch-dictionary $\mathbf{Y}$ by cropping them into patches, and solve Lasso problem Eq. (3), and then update $\boldsymbol{\alpha}$ by the process of occlusion prediction scheme (see Algorithm 2)
4: Add the most optimal candidate in the $(t-1)$-th frame into template set $\mathbf{T}$ and update it (see Algorithm 3)
5: Adopt the alignment-pooling algorithm to ascertain the most optimal candidate

Ensure:

Tracking results of each frame

3.2. Occlusion patch location prediction

An occlusion prediction scheme is adopted to predict occlusion patch location of current tracking rectangular box, and the prediction results are regarded as the occlusion prediction results of M candidates surrounding with the current tracking rectangular box. In practice, we keep the corresponding contribution factor to be constant 1 for those occluded patches and increase the contribution factor for those non-occluded patches. Equivalently, we rewrite Eq. (8) as

$$\max_j \sum_{P_i \in P^{unocc}} \alpha_i^j v_{ii}^j + \sum_{P_i \in P \backslash P^{unocc}} \alpha_i^j v_{ii}^j, \quad (9)$$

where P_i denotes the location number of patch i, P means the patches set which is assembled by $\{P_1, P_2, \ldots, P_N\}$, and P^{unocc} means the non-occluded patches, so $P \backslash P^{unocc}$ means the occluded patches.

Inspired by the classification method [2], the occlusion prediction scheme, which uses patches as instances and then classify these patches by the designing patch trainer, is proposed. First, the center point of the object target is taken. And then the parameter R, which is larger than the height of the labeled bounding box, is used to set the range of the circle searching area. After the positive and negative patches in this circle are labeled, support vector machine (SVM) is adopted to design a patch classifier. With this classifier, positive and negative patches within the most optimal candidate bounding box region can be classified and then the total negative patches number, represented by n_n, can be computed. Therefore, we use the following formula to predict occlusion case of one candidate:

$$r_{occ} = \frac{n_n}{N}, \quad (10)$$

where r_{occ} is the occlusion ratio, which reflects the occlusion degree. Furthermore, we can get the occlusion position in the rectangular box. In this paper, the white pixel is used to indicate those non-occluded regions, which is shown in (3).

Algorithm 2. Occlusion prediction scheme.

Require:

The tracking result in frame t, $t \geq n+1$ and M candidates surrounding with frame t, which has been divided into N patches; And then use patch classifier, which is formed by combining SVM [28,29] and MIL [2], to judge the occluded patches i in frame t and update their contribution factors according to the computation formula of α_i; And finally obtain $\boldsymbol{\alpha}$ and occlusion ratio r_{occ} for frame t;

Ensure:

Contribution factor $\boldsymbol{\alpha}$ of the M candidates and occlusion ratio r_{occ} for frame t.

3.3. Template update

In this paper, the template update algorithm in [14] is adopted, which introduces subspace learning into sparse representation to make the templates adapt to the appearance variation of the object target and proposes a balance scheme between the old and the new templates. This template update algorithm is described in Algorithm 3.

In detail, the selection of discarded templates comply with the rationale that the earlier tracking results are more accurate and should be stored longer than new obtained tracking results. Therefore, firstly, a cumulative probability sequence is generated, which is denoted as $\{0, 1/(2^{n-1}-1), 3/(2^{n-1}-1), 7/(2^{n-1}-1), \ldots, 1\}$, and each element of which means the update probability from the first template to the fiftieth template. Secondly, a random number r is generated according to uniform distribution on the unit interval [0,1]. Finally, the discarded template

can be chosen and then be replaced by determining which section of the sequence the random number lies in. Therefore, the latter in the template sequence are likely to be replaced than the former from the perspective of probability.

Algorithm 3. Template update.

Require:
Observation vector of target estimation **p**, eigenbasis vectors **U**, template set **T**, regularization parameter λ and occlusion ratio r_{occ};
Generate a sequence of number in ascending order and normalize them into [0,1] as the probability for template update;
Generate a random number between 0 and 1 which is for the selection of which template to be discarded;
Solve and obtain **q** and **e**;
If the current frame number can be divided by 5, add $\mathbf{p}=\mathbf{Uq}$ to the end of the template set **T**;

Ensure:
New template set **T**.

4. Experiments and evaluations

Our RC-PFRSR method is implemented in MATLAB and runs at around 1.5 frames per second on a PC with an Intel 3.6 GHz Dual Core CPU and 8GB memory. We manually label the location of the object target in first frame for each sequence and set the regularization constant $\lambda=0.01$, $n=15$, $N=9$, and $d=256$ in our experiments. According to the feedback of our experiments, for *woman* and *stone* sequences, we resize the object target image patch to 36×36 pixels and extract overlapped 12×12 local patches within the object target region with 0 pixel as step length. While, we adopt the same dividing patches way as [14] for other eleven sequences.

We evaluate the performance of our RC-PFRSR method on thirteen challenging sequences, which includes *Caviar*1, *Caviar*2, *Caviar*3, *Faceocc*1, *Faceocc*2, *Singer*, *Board*, *Woman*, *Face*, *Stone*, *Davidin*300, *Car*4, *Car*11. In order to make the comparisons persuasive, we obtain the results of six state-of-the-art methods by running the source codes provided by the authors with same parameters or copying the experimental results from the published papers if the source codes are not provided.

It is worth noting that, in tracking scenario, noise is embodied by different factors, such as background clutter, illumination and occlusion. If one region is not occluded, it may also happen to the common drifting phenomenon due to the background clutter or illumination. Therefore, our method, which makes the useful patch information contribute more powers, can avoid or reduce the drifting causing by these factors.

4.1. Quantitative evaluation

We employ two evaluation criteria, the position center errors (in pixels) and overlap rate [14,30], to quantitatively

Table 1
Experimental comparison results in terms of position center errors (in pixels).

	IVT	ℓ1	PN	VTD	MIL	ASLSA	Ours_best
Faceocc2	10.2	11.1	18.6	10.4	14.1	3.8	**3.1**
Woman	167.5	131.6	9.0	136.6	122.4	2.8	**2.3**
Board	165.4	177.0	97.3	96.1	60.1	7.3	**6.8**
Singer	8.5	4.6	32.7	4.1	15.2	4.8	**2.3**
Car11	2.1	33.3	25.1	27.1	43.5	2.0	**2.0**
Stone	2.2	19.2	8.0	31.4	32.3	1.8	**1.1**

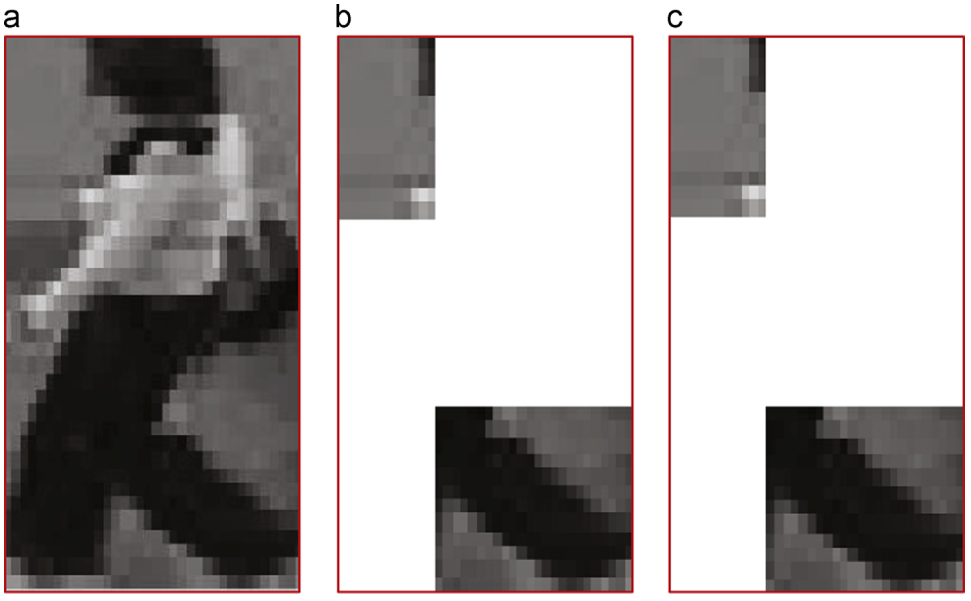

Fig. 3. One example of the occlusion case. This picture comes from the 16-th frame of *woman* sequence where (a) means the original image, (b) means the occlusion case by adopting label method. (c) means the occlusion case by adopting probability method. And the white pixel means the non-occluded part for image (a).

Table 2
Experimental comparison results in terms of overlap rate.

	IVT	$\ell 1$	PN	VTD	MIL	ASLSA	Ours_best
Faceocc2	0.59	0.67	0.49	0.59	0.61	0.82	**0.84**
Woman	0.19	0.18	0.60	0.15	0.16	0.78	**0.82**
Board	0.17	0.15	0.31	0.36	0.51	0.74	**0.75**
Singer	0.66	0.70	0.41	0.79	0.33	0.81	**0.88**
Car11	0.81	0.44	0.38	0.43	0.17	0.81	**0.82**
Stone	0.66	0.29	0.41	0.42	0.32	0.56	**0.68**

Table 3
Experimental comparison results in terms of position center errors for two methods.

	ASLSA	Ours_best
DavidIndoor	4.19	**3.18**
Faceocc1	23.25	**4.53**
Caviar1	1.44	**1.15**
Caviar2	1.85	**1.35**
Caviar3	5.06	**2.38**
Face	11.53	**10.39**
Car4	3.84	**3.40**

Table 4
Experimental comparison results in terms of overlap rate for two methods.

	ASLSA	Ours_best
DavidIndoor	0.76	**0.77**
Faceocc1	0.74	**0.92**
Caviar1	0.90	**0.91**
Caviar2	0.84	**0.88**
Caviar3	0.74	**0.87**
Face	0.73	**0.77**
Car4	0.90	**0.91**

evaluate the performance of our RC-PFRSR tracking method. In this paper, we tend to use overlap rate to evaluate our method because of its robustness. First, tested on *Caviar*1, *Caviar*2, *Faceocc*1, *Singer*, *Car*4, *Car*11, *Faceocc*2, *Stone* sequences, we compare our RC-PFRSR method with six state-of-the-art tracking methods, i.e., the incremental learning visual tracking (IVT) method [8], the ℓ_1 tracker [12], the multiple instance learning (MIL) tracker [2], the visual tracking decomposition (VTD) method [6], the P-N learning (PN) tracker [3] and ASLSA. The comparison result is shown in Tables 1 and 2. Then, tested on *Board*, *Woman*, *Face*, *Caviar*3 sequences, we compare our RC-PFRSR with ASLSA. This comparison result is shown in Tables 3 and 4. In addition, Fig. 4 shows the overlap case between the ground truthes and the tracking results, which are compared with six state-of-the-art tracking methods, while Fig. 5 shows the comparisons only between our RC-PFRSR method with ASLSA. Therefore, RC-PFRSR method can well perform the effective tracking results.

4.2. Qualitative evaluation

In the experiments, our RC-PFRSR method can obtain superior tracking results in many sequences. For simplicity, we make qualitative evaluations in the following cases.

Background clutter case: Fig. 6 represents the tracking results of the *Face* and *Stone* sequences with the heavy background clutters. For *Face* sequence, we compare RC-PFRSR method with ASLSA method, which shows that RC-PFRSR method outperforms RC-PFRSR method, such as at frame 155 in Fig. 6(a). For *Stone* sequence, we compare RC-PFRSR method with six state-of-the-art methods, where MIL and VTD methods start to be way off the correct location since 394 frame shown in Fig. 6(b). In general, for these two sequences, only ASLSA method can get an approximated tracking result with the corresponding

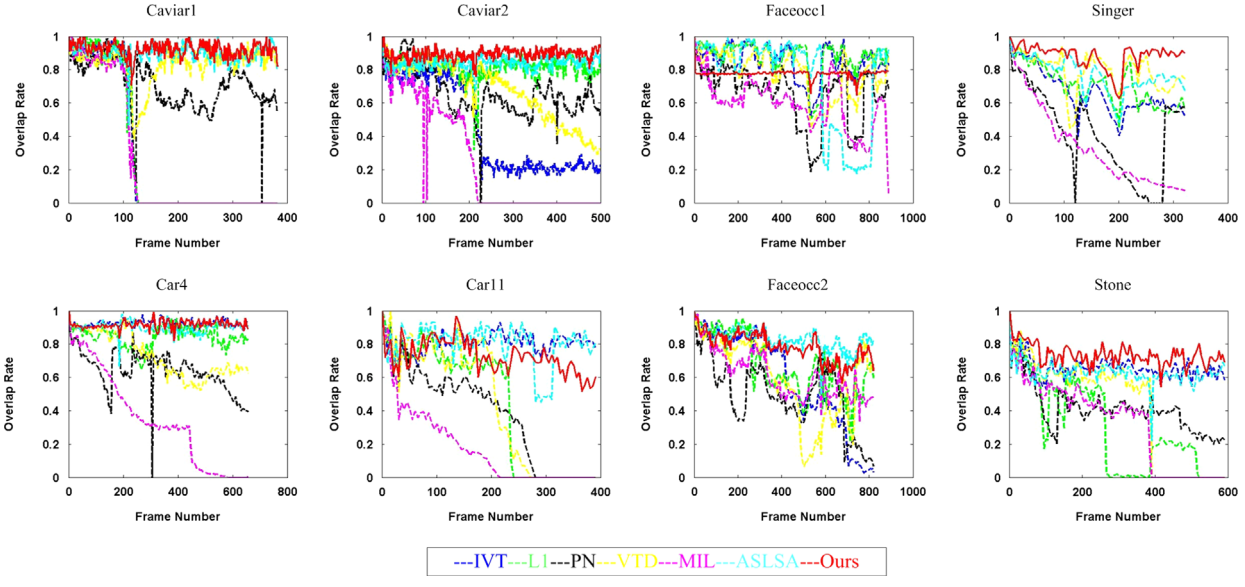

Fig. 4. Quantitative evaluation of the trackers in terms of overlap rate by comparing our RC-PFRSR method with ASLSA on eight different datasets.

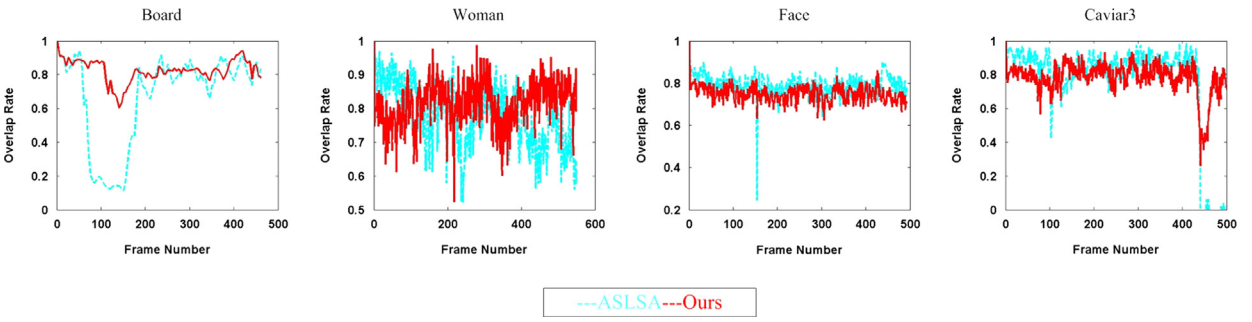

Fig. 5. Quantitative evaluation of the trackers in terms of overlap rate by comparing our RC-PFRSR method with ASLSA on four different datasets.

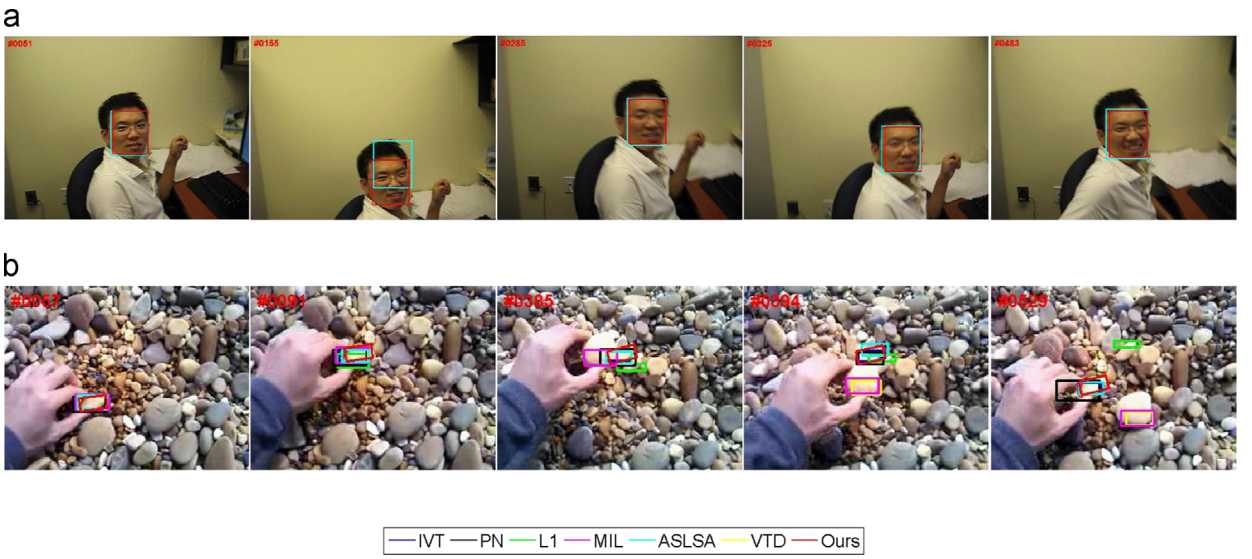

Fig. 6. Tracking results when the object targets happen to background clutters. Our RC-PFRSR method is compared with ASLSA in (a) and compared with six state-of-the-art tracking methods in (b).

ground truthes. However, our RC-PFRSR method can get the better and robust tracking result with the lowest tracking error.

Occlusion case: Fig. 7 demonstrates how accurate and robust our method is when the object target undergoes a heavy or long-time partial occlusion. In detail, from *Caviar*1 sequence, we can see that IVT, PN, L1, MIL, VTD methods start to deviate from the correct location since 122 frame. From *Caviar*2 sequence, we can see that MIL method starts to deviate from the correct location and only our method can have a more accurate tracking location since 215 frame. From *Caviar*3 sequence, we can see that only ASLSA method and our method can have a more accurate tracking location since 82 frame, and ASLSA method is also way off the correct location at 440 frame unfortunately. In conclusion, our method obviously has the better tracking results than those six state-of-the-art methods.

Illumination case: Fig. 8 represents the tracking results in *Car*4, *DavidIndoor*, *Car*11 sequences with the large illumination variation. For *Car*4 sequence, only PN and MIL methods happen to some deviations from the correct location in 146 and 653 frame respectively. For *Car*11 sequence, four state-of-the-art methods gradually start to be way off the correct location except IVT and our method can keep the accurate tracking since 188 frame. Generally speaking, our method has a more accurate tracking result and achieves a lower tracking error.

5. Conclusion

In this paper, we propose the RC-PFRSR method to assign a larger contribution factor on the more discriminative patches. In detail, we design a contribution factor matrix α to modify the patch coefficients with occlusion and keep the patch coefficients without occlusion. That is, the interesting information, got by occlusion prediction scheme, helps to reduce the drifting. It is worth noting that our method cannot only improve the tracking results in datasets with occlusions but also have the superior tracking results in datasets with background clutters and illumination changes because of its occlusion prediction scheme, which can discriminate the useful information. The experiment results

Fig. 7. Tracking results when the object targets happen to occlusions.

Fig. 8. Tracking results when the object targets happen to illumination changes. Our RC-PFRSR method is compared with ASLSA in (b) and compared with six state-of-the-art tracking methods in (a) and (c).

on benchmark datasets demonstrate that our method is effective and outperforms the state-of-the-art methods.

Acknowledgments

This study was supported by the Key Foundation Research Project of Shenzhen (No. JC201104210033A), Innovation Project of Scholars from Overseas of Shenzhen (KQCX-20120801104656658), the Technology Innovation Project of Shenzhen (No. CXZZ20120618155717337, CXZZ2013031-8162826126), Faculty Research Grant of Hong Kong Baptist University (No. FRG2/12-13/082), and the grant of National Natural Science Foundation of China (No. 61272366). The authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

References

[1] H. Grabner, H. Bischof, On-line boosting and vision, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 260–267.
[2] B. Babenko, M.-H. Yang, S. Belongie, Visual tracking with online multiple instance learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 983–990.
[3] Z. Kalal, J. Matas, K. Mikolajczyk, Positive and Negative learning: bootstrapping binary classifiers by structural constraints, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 49–56.
[4] S. Wang, H. Lu, F. Yang, M.-H. Yang, Superpixel tracking, in: International Conference on Computer Vision, 2011, pp. 1323–1330.
[5] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 798–805.
[6] J. Kwon, K. M. Lee, Visual tracking decomposition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1269–1276.
[7] L. Matthews, T. Ishikawa, S. Baker, The template update problem, IEEE Trans. Pattern Anal. Mach. Intell. 26 (6) (2004) 810–815.
[8] D.A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, IEEE Trans. Int. J. Comput. Vis. 77 (1–3) (2008) 125–141.
[9] E.J. Candes, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Commun. Pure Appl. Math. 59 (8) (2006) 1207–1223.
[10] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.
[11] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[12] X. Mei, H. Ling, Robust visual tracking using l1 minimization, in: International Conference Computer Vision, 2009, pp. 1436–1443.
[13] X. Mei, H. Ling, Y. Wu, E. Blasch, L. Bai, Minimum error bounded efficient l1 tracker with occlusion detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1257–1264.
[14] X. Jia, H. Lu, M.-H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1822–1829.
[15] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, C. Kulikowski, Robust and fast collaborative tracking with two stage sparse optimization, in: IEEE Conference on Computer Vision, 2010, pp. 624–637.
[16] W. Zhong, H. Lu, M.-H. Yang, Robust object tracking via sparsity-based collaborative model, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1838–1845.
[17] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (Methodol.) (1996) 267–288.
[18] D.L. Donoho, For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution, Commun. Pure Appl. Math. 59 (6) (2006) 797–829.
[19] E.J. Candes, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inf. Theory 52 (12) (2006) 5406–5425.
[20] C. Hong, N. Li, M. Song, J. Bu, C. Chen, A level-set based tracking approach for surveillance video with fusion and occlusion, in: 2010 Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT), 2010, pp. 156–161.
[21] W.M Yu J, D. Tao, Adaptive hypergraph learning and its application in image classification, IEEE Trans. Image Process. 21 (7) (2012) 3262–3272.
[22] C. Hong, J. Yu, X. Chen, Image-based 3d human pose recovery with locality sensitive sparse retrieval, in: IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2013, pp. 2103–2108.
[23] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (5) (2014) 2019–2032.
[24] W.M. Yu J, R. Hong, Image clustering based on sparse patch alignment framework, Pattern Recognit. 47 (11) (2014) 3512–3519.
[25] J.A. Tropp, S.J. Wright, Computational methods for sparse solution of linear inverse problems, Proc. IEEE 98 (6) (2010) 948–958.
[26] C. Bao, Y. Wu, H. Ling, H. Ji, Real time robust l1 tracker using accelerated proximal gradient approach, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1830–1837.
[27] D. Wang, H. Lu, M.-H. Yang, Online object tracking with sparse prototypes, IEEE Trans. Image Process. 22 (1) (2013) 314–325.
[28] Y. Bai, M. Tang, Robust tracking via weakly supervised ranking Supported Vector Machine, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 1854–1861.
[29] S. Avidan, Support vector tracking, IEEE Trans. Pattern Anal. Mach. Intell. 26 (8) (2004) 1064–1072.
[30] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (Visual Object Classes) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.