Signal Processing ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Signal Processing

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

A robust local sparse tracker with global consistency constraint

Xinhua You [a,b], Xin Li [b], Zhenyu He [b,*], X.F. Zhang [b]

[a] Zhixing College of HuBei University, China
[b] School of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, China

ARTICLE INFO

Article history:
Received 12 May 2014
Received in revised form
13 September 2014
Accepted 17 September 2014

Keywords:
Visual object tracking
Global consistency
Local sparse model
Template update
Partial and spatial information

ABSTRACT

In the field of visual object tracking, partial occlusion and the variation of illumination, pose and background are the core problems to be handled. More and more visual tracking methods tend to exploit part or local features to deal with the above problems. However, single local features may lead to overfitting and drifting problem, as will cause the failure of tracking task. In this paper, we propose a novel tracking method by exploiting the partial and spatial information with a global regulation on the stabilization of local features. With the local features and the global constraint, the problems of occlusion and variation can be well solved and a stable performance can be obtained without overfitting. In the first stage, overlapped patches are used to hold the local features and each patch is reconstructed with all the template patches. The reconstruction coefficients are obtained by solving the ℓ_1 regularized least square problem. In the second stage, a global constraint is added to find the final result. The constraint is achieved by restraining the difference in contributions of each patch. Additionally, we employ occlusion information to improve the template update strategy. The experiment results on several widely used benchmark datasets demonstrate that our method is effective and outperforms the state-of-the-art trackers.

1. Introduction

The task of visual object tracking is to find the location of the target frame by frame. Applications of object tracking spread over video processing and human–computer interaction, such as vehicle navigation, automatic drive, video surveillance and intelligent robots. Despite the progress in theory and the improvement in experimental results, visual object tracking remains a challenging task because of the destabilizing factors, such as the changes of illumination, complicated background, occlusions in the video sequence. Occlusion and drifting often occur in tracking, and a variety of approaches have been raised to overcome these problems.

A series of tracking methods focus on the differences between the target and the background, which are called as the discriminative method. The discriminative method usually trains a classifier on the templates and then classifies each candidate with a probability. And the candidate with the maximum probability is selected as the tracking result. Avidan [2] first casts tracking as a binary classification problem and trained an ensemble of weak classifiers on-line to distinguish the object from the background. In his method, the combined classifier is used to classify each pixel in the frame as the object or the background, which gives a confidence map. Later, Grabner and colleagues [3] proposed an on-line semi-supervised boosting method to alleviate the drifting problem. They do

* Corresponding author.
E-mail address: zyhe@hitsz.edu.cn (Z. He).

http://dx.doi.org/10.1016/j.sigpro.2014.09.019
0165-1684/

the update process by combining a given prior and an online classifier. These classifiers may be degraded by the incorrectly labeled training samples. To train the samples more accurately, Babenko and colleagues [4] exploited the multiple instance learning (MIL) instead of traditional supervised learning, which can reduce the dependency on the tagging accuracy and gives a more robust tracker with fewer parameters. Kalal and colleagues [5] improved the binary classifier by the processing of structured unlabeled data. The learning process of their method is guided by positive and negative constraints to get a better performance. Wang and colleagues [6] used superpixel as the image representation unit from the perspective of mid-level vision. Then, they presented a discriminative appearance model based on the superpixels to distinguish the target and the background.

Another kind of tracking methods is the generative method, which usually uses the templates to represent the candidate. The processing unit can be pixel, patch or part of the object. Adam and colleagues [7] used multiple image fragments to represent the template object and each fragment votes on the possible positions and scales of the object by comparing their image fragment histogram. The fragment format facilitates the handling of the occlusion problem. Kwon and colleagues [8] proposed a visual tracking decomposition scheme for observation and motion models. The observation model is decomposed into multiple basic models constructed by sparse principal component analysis (SPCA) of feature templates. Then, several trackers are used and each one of them takes charge of certain change of the object to get a better performance. Ross and colleagues [9] presented a dynamical tracking method, which incrementally learns a low-dimensional subspace representation to deal with the variation in the appearance of the target. Besides these generative methods that focus on finding a powerful representation unit, sparse representation is also introduced into generative models.

Mei and Ling [10] introduced the sparse theory into tracking problem on a particle filter framework. And they added trivial templates to deal with the occlusion and corruption problem. The sparsity is achieved by solving an ℓ_1-regularized least squares problem. Later, another ℓ_1 tracker with minimum error bound and occlusion detection is proposed in [11]. The minimum error bound is calculated by a linear least squares equation and serves as a guide for particle resampling in a particle filter framework. The occlusions are detected by investigating the trivial coefficients in the ℓ_1 minimization. In [12], Liu and colleagues proposed a local sparse appearance model and adopted a representation-based voting map for tracking. Another local sparse tracking is proposed by Jia and colleagues [1]. They developed a robust tracking method based on the structural local sparse appearance model and used an alignment-pooling method to obtain partial and spatial information of the target. These methods focus on the usage of local features and structure which perform well in dealing with occlusion and variation. However, single local features may lead to overfitting and drifting problems.

Local sparse based methods can handle the partial occlusion and variation of target well. However, local features may be misled by the similar background because it is a common case that a patch on the background is similar to some patches of the target. In this case, the tracking result may contain deviation and consequently drifting occurs. Therefore, the global constraint should be added to the local methods to alleviate the drifting problems. In this paper, we propose a local sparse model with a global constraint. The constraint is achieved by giving a penalty to the contribution difference of each patch on the same candidate. We use patch as the

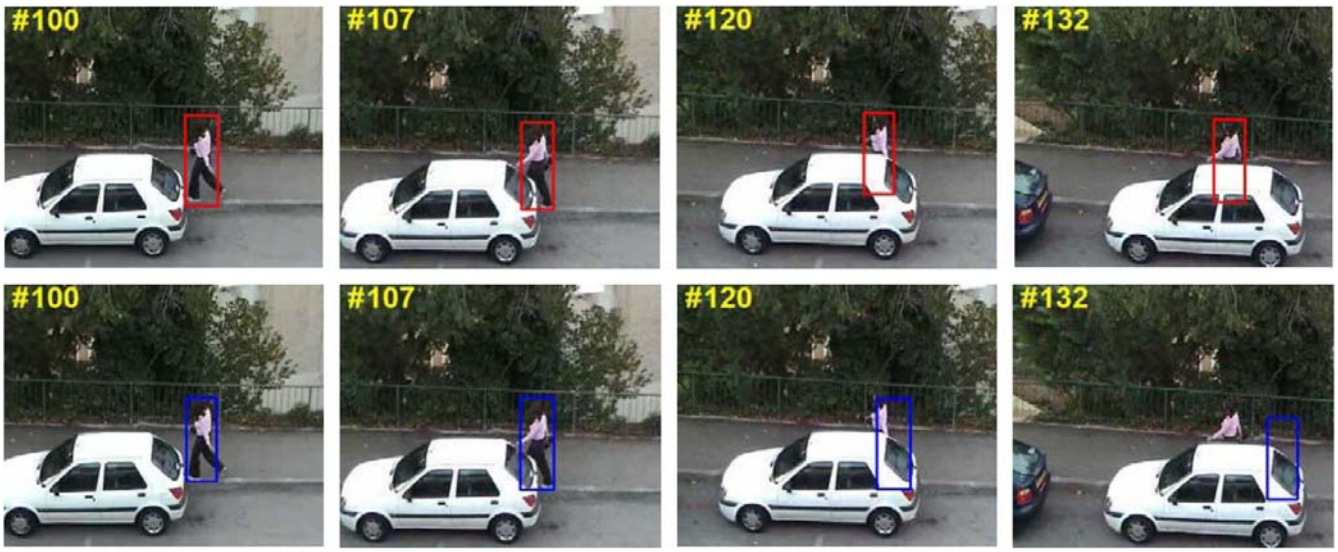

Fig. 1. Result comparison between the local sparse based tracking method and the proposed tracking method. The pictures are got from WomanSequence dataset [1]. Our results are shown in the first row, while the results of the local sparse tracking method are shown in the second row. The number in the upper left corner is the frame number. It is easy to see the blue results occur a drifting in frame 120 and 132 because the local sparse based tracking method is misled by the background region that is similar to the woman, as is the black windows of the car in this figure. However, the proposed method overcomes the drifting problem and obtains promising results.

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019

processing unit to keep the partial information and sample the patches in a fixed order both on the candidate samples and the template targets, which keeps the spatial information. Then, each candidate patch gets a score by solving the ℓ_1 regularized least squares problem. The global constraint achieves by giving a bigger penalty to the candidate with bigger variance of all the patches' confidence. Additionally, the distribution of the patches' score indicates the occlusion information, which can be used to guide the template update. We test the proposed method on several benchmark datasets, and the promising results demonstrate that the proposed method is effective and outperforms the state-of-the-art trackers. An illustrative example of our method can be found in Fig. 1.

Summarily, our contribution is threefold. The first contribution is proposing a novel local sparse model with a global constraint, which can reduce the drifting problems. The second contribution is giving a two-stage algorithm to exploit the partial and spatial information without overfitting problems. The third contribution is raising exploiting occlusion information to guide the template update, which can improve the template update strategy.

The rest of this paper is organized as follows. Section 2 is a short review of related research. Section 3 describes our local sparse method with a global constraint. Section 4 discusses the template update method with occlusion detection. Section 5 gives a tracking framework. Section 6 shows the experimental details. A short conclusion of our work is drawn in Section 7.

2. Related work

Some trackers [13,14] focus on realtime tracking, while others focus on tracking accuracy. In this paper, we focus on dealing with the drifting problem. Essentially, single object tracking is an image classification problem which can be solved with a variety of learning methods [15–18]. These methods are all discriminative models, and our model is based on the generative model with spare representation. Sparse representation has natural superiority on image processing area, such as multimodal sparse coding [19] and image clustering [20]. Many sparse trackers [10–12,21] have been proposed for tracking. Mei and Ling [10] introduced the sparse representation into the tracking system and regarded tracking as a sparse approximation problem in a particle filter framework. They achieved the sparsity by solving an ℓ_1-regularized least squares problem with nonnegativity constraints and handled the occlusion problem by adding trivial templates. In order to deal with occlusion and variation more flexibly, Jia and colleagues [1] presented a structural local sparse appearance model. This structural local model exploits partial and spatial information of the target. And they introduced the incremental subspace learning into sparse representation to improve the template update, which can alleviate the probability of drifting. However, the local features and the local sparse are prone to causing overfitting and drifting problems. In our method, we improve the local sparse model in [1] by adding a global constraint to get a novel model that keeps the structural and the spatial information. As a result, our model can handle the overfitting and drifting problem effectively because it takes the global distribution of the local results into consideration and gets the result based on the combined information of local and global.

Later, Mei et al. [11] proposed an efficient ℓ_1 tracker with minimum error bound and occlusion detection. They calculated the minimum error bound from a linear least squares equation first. The error bound is used to guide for particle resampling in a particle filter framework, which can remove the insignificant samples. The occlusion is detected by investigating the trivial coefficients, which is got from the same ℓ_1 minimization. In our algorithm, we use the occlusion information to guide the template update procedure. However, the occlusion information in our model is got in a new method. The patch is sampled in a fixed order both on the template and the samples, and each patch is sparsely represented by the template patches. Then, a score can be obtained of each patch independently and the score distribution is used to detect occlusion. Therefore, our occlusion information is more accurate and the inaccurate estimate of one patch will not influence other patches, which is more robust to estimated deviations. In order to get a faster sparse tracker, Xiao et al. [21] achieved the sparsity by solving ℓ_2-regularized least square problems. And the appearance of the tracked target is modeled with PCA basis vectors and square templates. However, the speed of our tracker is about the same as theirs.

3. Local sparse with global constraint model

In this section, we first give the framework of object tracking and then review a local sparse method proposed in [1], which can keep the partial and spatial features but have drifting problems. Afterwards, we analyze the overfitting problem of the local method with experiments and

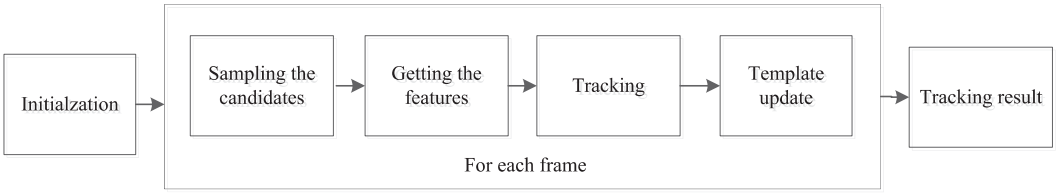

Fig. 2. The framework of object tracking. The first stage is initialization. The location of the target is given and the initialization parameters, such as the number of templates, the size of the resized image and the number of patches, are set. The second stage is tracking, which includes sampling the candidates, getting the features of the template and the candidate, selecting the most similar candidate and updating the template set. The third stage is outputting the result.

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019

show the problem with visible figures. Later, we give a stable and robust model with a global constraint and develop a two-stage algorithm to solve the proposed formulation.

The framework of visual object tracking can be divided into three stages: initial stage, tracking stage and result output stage, which can be seen in Fig. 2.

3.1. Local sparse method

In visual object tracking field, motion model and appearance model are used to capture the features of the target object. In our method, motion information is exploited in the candidate sample stage. With the motion information, the particle filter can decide which area has a greater probability of containing the target. The appearance model usually includes the features of color, shape and texture. However, in the tracking framework of sparse representation, gray values of an image are directly used without a feature extraction procedure. In our method, we also use the gray value as the feature.

An image patch is exploited as the processing unit in this model for its advantage in dealing with occlusion and variation. The shape of the patch can be a square or a rectangle, which contains dozens of pixels. In order to exploit the spatial information, we sample the patches in a fixed order. First, the original template or candidate image is resized as a specified size. Second, the image patches are sampled with the same size in the same order, which is from top to down and from left to right. In order to make one patch have a larger probability to contain a complete part of the target object, overlapped patches are sampled both on the templates and the candidate samples. Therefore, the position information of each patch can be obtained.

Given the template set $\mathcal{T}=\{t_1,t_2,\ldots,t_n\}$, patches are sampled with the same spatial layout on the resized target region. The jth patch of the ith template is denoted as p_j^i, where $i\in\{1,2,\ldots,n\}$ and $j\in\{1,2,\ldots,m\}$, m is the total patch number of one template and n is the template number. Then, the dictionary of the template patch can be denoted as $\mathcal{D}=[d_1,d_2,\ldots,d_{n\times m}]\in\mathbb{R}^{l\times n\times m}$, where l is the dimension of the image patch vector. The image patch vector d in $\mathcal{D}$ is got from the templates and each vector d is processed with ℓ_2 normalization. The matrix form of a candidate is denoted as $\mathcal{Z}=[z_1,z_2,\ldots,z_m]\in\mathbb{R}^{l\times m}$. With the sparsity assumption, each patch of the candidate z_i can be linearly represented by a few elements of the dictionary $\mathcal{D}$. And the coefficients can be obtained by solving the classical ℓ_1-regularized least squares equation

$$\arg\min_{a_i} \quad \|z_j-\mathcal{D}a_j\|_2^2+\lambda\|a_j\|_1,$$

$$\text{s.t.} \quad a_{j,k}\geq 0, \quad k\in\{1,2,\ldots,n\times m\} \tag{1}$$

where $\|.\|_1$ and $\|.\|_2$ denote the ℓ_1 and ℓ_2 norms, respectively. a_j is the coefficient vector, and $a_{j,k}\geq 0$ denotes each element in a_j is nonnegative, which satisfies the nonnegative constraint. With Eq. (1), each candidate patch z_j can get a corresponding sparse coefficient vector $\hat{a}\in\mathbb{R}^{M\times 1}$, where $M=m\times n$ denotes the total template number. As each patch is sampled from a certain fixed part of the target object and is linearly represented by $\mathcal{D}$ with all the patches, the template patch in $\mathcal{D}$ with the same position of the candidate patch will obtain larger coefficients. These coefficients measure the similarity between the template patches and the target patch. Therefore, the confidence of

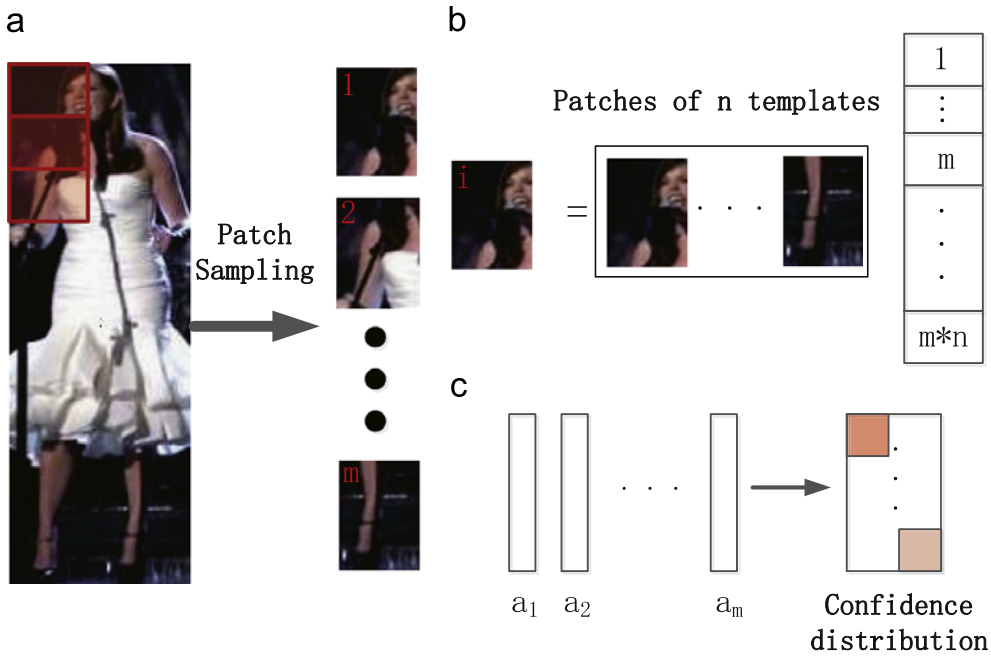

Fig. 3. An illustration of the local sparse method. First, sample the template and candidate images with overlapped patch in the same spatial layout manner, which is shown in (a). (b) Each candidate patch is linearly represented by all the template patches. (c) The ℓ_1-regularized least squares solutions of each patch from a_1 to a_m. And with these coefficients, a confidence distribution of a candidate can be obtained. The confidence of each candidate patch is given in Eq. (2).

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019

each patch to be the real target patch can be calculate as

$$C_j = \sum_{q=1}^{n-1} a_{jj+m\times q}, \tag{2}$$

where $j+j\times q$ is the patch index, whose position is the same as patch j. C_j denotes the confidence of patch j. The procedure of the local sparse method is shown in Fig. 3. With the local sparse method, the confidence of each patch with a fixed part of the target can be obtained and, with all the fixed parts, we can get the confidence distribution of the whole target.

3.2. Global constraint model

With the local sparse method, the confidence of each patch to be the real target can be obtained. However, simply selecting the candidate with the highest confidence sum of all the patches may lead to overfitting and drifting problems, which can be seen in Fig. 4. The reason of overfitting is that some local features of the background have a great similarity with the target patch, such as in Fig. 4(a). In this case, the sample candidate which contains a similar background will have a high confidence sum and its confidence may be larger than the accuracy candidate. With a lot of experiments, we find that wrong or inaccurate candidates with a high confidence usually includes a few high confidence patches, but other patches' confidences are very low, which can be seen in Fig. 4(c). Therefore, a constraint should be added on a global manner to prevent the overfitting problem. Here we put a constraint on the confidence distribution by giving a penalty to the candidate with a large variance. With Eq. (1), we can get the confidence of each patch, and we set the cost of the ith patch to the real patch as

$$\mathcal{C}_j = -C_j. \tag{3}$$

We denoted the candidate set as $\mathcal{Y}=\{y_1,y_2,\ldots,y_Q\}$, where Q is the number of candidate samples. And the total cost of candidate y_j to be the real target can be written as the sum of the $\mathcal{C}_j$. With the distribution constraint, the objective function can be obtained as

$$\arg\min_{y_j} \sum_{j=1}^{m} \mathcal{C}_j + \mu \sum_{i=1}^{m} \|\mathcal{C}_j - \overline{\mathcal{C}}_j\|_2^2 \mathcal{F}(\sigma_j), \tag{4}$$

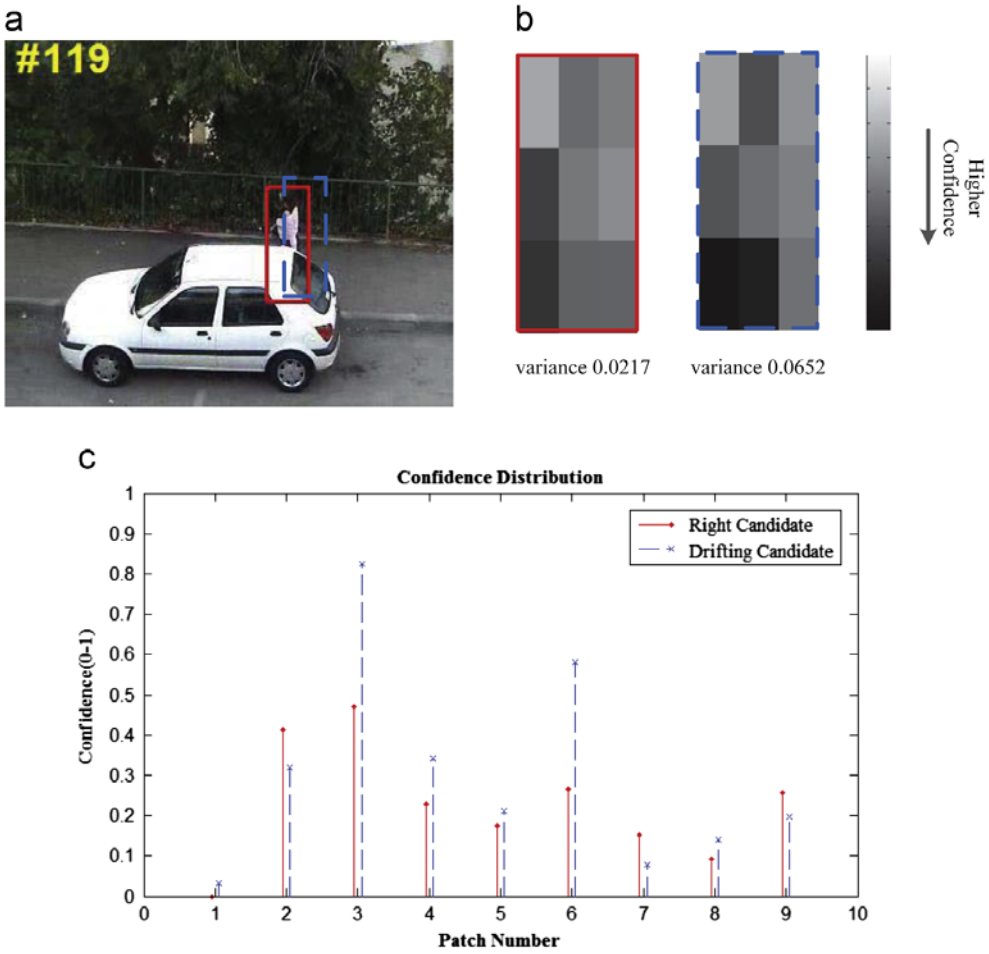

Fig. 4. An illustration of drifting with local sparse method and its confidence distribution. The picture in (a) is from dataset WomanSequence. The solid and dashed line bounding boxes represent our result and the local sparse result, respectively. With the local sparse method, the confidence of each patch in the two results can be got and its distribution is shown in (b) and (c). The left gray image in (b) shows the contributions of each local patch of the solid line bounding and the right one corresponding to the dashed line bounding box. We can see the dashed lien bounding box result is drifting and the reason is that the color of the car window is very similar to the trousers' of the target. From (b) and (c), it is easy to see part of the blue result occurs overfitting and only a few local patches have a high confidence but others are not.

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019

where m is the total patch number of one target sample, $\overline{C}_j$ is the mean of C_j, μ is an adjustment parameter, $\mathcal{F}(\sigma_j)$ is the distribution of the contribution of the patches, σ_j is the distance of patch j to the center of the target. We can see that the second part of Eq. (4) is a weighted variance of C_j and $\mathcal{F}(\sigma_j)$ is the weight distribution. As we can see in Fig. 3, the patches far away from the center of the target may contain the background and in the tracking procedure the background is changing. Hence the weight of a patch should decrease along with the increase of σ, just like the Gaussian distribution. This objective function covers two aspects. The first aspect is the solution to local sparse, which selects the most similar candidate sample based on the partial and spatial features. The second aspect takes the confidence distribution of each local patches into consideration, which can avoid the overfitting problems. In Fig. 4, the solid line bounding box has a higher total confidence than the dashed line one, however the variance of the dashed line bounding box is three times as that of the solid line one.

The value of adjustment parameter μ in Eq. (4) depends on the situation of dataset or even the background with the same dataset, and hence it is an empirical value, which is hard to get. In order to solve this problem, we exploit a two-step method to achieve the objective of local sparse with a global constraint. In the first step, we select the candidate samples with the cost $\mathcal{C}=\sum_{i=1}^{m} C_j$, which is less than an error bound eb and we get a candidate set T_{eb}. In the second step, we select the candidate sample with minimum variance as the tracking result from T_{eb}. Therefore, the objective function can be rewritten as

$$\begin{aligned}&\arg\min_{y_p} \sum_{j=1}^{m} \|C_j-\overline{C}_j\|_2^2\mathcal{F}(\sigma_j),\\ &\text{s.t.} \sum_{j=1}^{m} C_j \le eb\end{aligned} \tag{5}$$

where m is the total patch number of a candidate, y_p stands for the pth candidate sample and $p\in\{1,2,\ldots,Q\}$, C_j is the cost of the jth patch in the pth candidate, $\overline{C}_j$ is the mean cost of the patches in candidate y_p. With Eq. (5), we can get a candidate with a relative high confidence but a small variance of the confidence distribution as the tracking result. The high confidence ensures that the candidate is similar to the real target, and the small variance keeps the result away from the overfitting candidates. The error bound value eb can be set by

$$eb=\alpha(C_{max}-C_{min}), \tag{6}$$

where C_{max} and C_{min} denotes the maximum cost and the minimum cost among all the patches, respectively.

4. Template update

In visual object tracking literature, template set is exploited to maintain the feature information [22] of the target. The tracking problem in each frame is to find the most similar candidate sample based on the template set. In essence, the tracking problem is a classification problem, and the template set is the training or feature set, which directly influences the classification results. Insufficient template set or incomplete feature space may lead to inaccuracy results, especially for tracking problem, whose target is changing all the time. Fixed templates or features cannot deal the variation of the target, such as the changing of illumination and pose, hence the update of the template is needed. Before using the template update mechanism, a series of problems need to be considered. How to set the update frequency? High frequency will result in drifting problem and low frequency cannot deal with the variance. Which template should be abandoned and which result should be added?

To solve the above problems, many methods [9,10,1,23] have been proposed to improve the mechanism of template update. Ross and colleagues [9] proposed an incremental principal component analysis algorithm to update the sample mean. They gave different weights to each template and adjusted the weights with a forgetting factor. However, they assumed that the reconstruction to be a Gaussian distribution with a small variance, which may not handle the partial occlusion well. The earlier tracking results are more accurate than the later ones, for which deviation of each frame may spread and accumulate. With this condition, Jia and Lu [1] generated a cumulative probability sequence, which leads to a slow update for old templates and a quick update for new ones. However, they did not directly exploit the occlusion information to guide the template update strategy. In our method, we use the occlusion information obtained by an occlusion detection mechanism, to guide the template update.

We use the local patch as the processing unit and use the local sparse method to get the confidences of each local patch, which is detailed in Section 3. In each template update process, we need to first select a template to be discarded, and then add a new result to the template set, for the number of template sets is fixed. In the first step, we can select the template with severe occlusion to be discarded. The degree of occlusion $\mathcal{O}_i$ can be measured by the reconstruction error and the variance of confidence distribution Var_i

$$\mathcal{O}_i=\|y_i-\mathcal{D}A_i\|_2^2+\eta\, Var_i, \tag{7}$$

where $\|y_i-\mathcal{D}A_i\|_2^2$ is the reconstruction error of candidate y_i, η is a normalization parameter. $\mathcal{O}_i$ is obtained when its corresponding observation y is added to the template set. In the second step, the incremental method proposed in [9] is exploited, which can maintain the constant feature of the target. And we add the occlusion information into the frame of sparse represent and subspace learning [1] to process the new added template. With the occlusion information, the occluded or corrupted patch can be obtained in the local sparse procedure, which is denoted as e'. Whereas, in [1], the occluded or corrupted pixels are unknown. Then the estimated target y can be represented as

$$y=Vx+e'=[V \quad I], \tag{8}$$

where V is the matrix of eigenbasis vectors, x are the coefficients of V, and e' is the error vector. This equation is solved as an ℓ_1 regularized least squares problem as

$$\min_c \|b-Dc\|_2^2+\lambda\|c\|_1, \tag{9}$$

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019

where $\|.\|_1$ and $\|.\|_2$ denote the ℓ_1 and ℓ_2 norms, respectively, $D=[V\ I]$, $c=[x\ e']^T$ and λ is the regularization parameter. The template update algorithm is described in Algorithm 1.

Algorithm 1. Template update with occlusion information.

Require:
Old template set $\mathcal{T}_{t-1}=\{t_1,\ldots,t_K\}$ and their occlusion degree $\mathcal{O}_i$, tracking result $\hat{y}_t$, occlusion patches e' of $\hat{b}_t$, eigenbasis vectors V

Ensure:
New template set $\mathcal{T}_t$

1: Discard the t_k from $\mathcal{T}_{t-1}$ with the largest $\mathcal{O}$;
2: Solve Eq. (8) with occluded patches e', and obtain x;
3: $t_t=Vx$ and $\mathcal{T}_t=\mathcal{T}_t^{tmp}\cup t_t$.

5. Proposed tracking framework

We use the particle filter [24], which is a Bayesian sequential importance sampling technique, to do sampling as [10]. With a particle filter framework, it is convenient to estimate and propagate the posterior probability density function. The prediction of the target distribution in the frame t can be computed as

$$p(x_t|S_{1:t-1})=\int p(x_t|x_{t-1})p(x_{t-1}|S_{1:t-1})\,dx_{t-1}, \tag{10}$$

where x_t is the state variable of an object at time t, $S_{1:t}=\{S_1,S_2,\ldots,S_{t-1}\}$ denotes all the observations up to time $t-1$. At the tth frame, S_t can be got with the template update and $p(x_t|S_{1:t})$ is got by the Bayes rule

$$p(x_t|S_{1:t})=\frac{p(S_t|x_t)p(x_t|S_{1:t-1})}{p(S_t|Z_{1:t-1})}, \tag{11}$$

where $p(S_t|x_t)$ denotes the observation likelihood. In this framework, the probability $p(S_t|x_{t-1})$ measures the similarity between the templates and a target candidate, which is implemented with our local sparse with the global constraint model.

6. Experiments

Implementation details: The proposed algorithm is implemented in MATLAB on one PC with an Intel 2.9 GHz Dual Core CPU and 2GB memory. The ℓ_1 sparse minimization problem is solved by the SPAMS package [25] and the regularization constant λ is set to 0.01. In the initial stage, we use the decision tree method to track the first 10 frames, which is used to get the templates for the proposed method. The template number in our experiment is 10. At the sampling stage, we resize all the candidates and the templates as $(32,32)$ in pixels in order to keep unity for different datasets. The patch size is set to $(16,16)$ and the sampling step length is 8 pixels. Therefore, each target region is cut into nine overlapping patches. In the two-step method, the α is set to about 0.9. And in the tracking step, we set the sample number Q as 600, i.e., in each frame, the candidate set has 600 samples. Our tracker has been tested many times on many datasets and gives promising results.

Datesets: Eight challenging sequences with different challenges are exploited to evaluate our tracking system. These sequences are Faceocc2 [7], Caviar, Woman, Car11, DavidIndoor [4], Singer, Stone and Board [26]. These video sequences cover the four main challenges of object tracking, which are partial and full occlusion, the variation of illumination, pose and scale variation and confused background, respectively. Each of the test datasets has its own focus of these challenges. Faceocc2 focuses on partial and full occlusion, the target of which is a people's head. In the sequence, the man shakes his head and uses a book and a hat to generate occlusion situation. Caviar and Woman focus on complicated occlusion and variation of pose, scale and background, the target of which is a pedestrian. Car11 and Stone focus on the confused background, the targets of which are moving objects. DavidIndoor and Girl datasets focus on the illumination and posture changes, the targets of which are human beings with a moving view.

Trackers: In order to estimate the performance of our tracking algorithm, we use seven state-of-the-art trackers with the same initial position of the tracking target for comparison. These tracking algorithms include the fragment-based (FragTrack) tracking methods [7], ℓ_1 tracker [10], PCA tracker [9], multiple instance learning (MIL) tracker [4], P–N learning (PN) tracker [5], visual tracking decomposition (VTD) method [8] and the ASLSA algorithm [1]. Some of their results are got by running the trackers with the source codes provided by the authors with the adjusted parameters, others are got from their papers or websites. In order to show the results more objectively, we also run the ASLSA tracker and our tracker 10 times and calculate the average results to do comparison, which shows the improvement of our tracker.

Evaluation: In visual object tracking, two metrics are widely exploited to evaluate the trackers' performance. The first metrics is the relative center position error (CPE) in pixels. The results are got by computing the distance between the center position of the tracking result and

Table 1
The center position error in pixels(CPE) comparing with other seven trackers on seven sequences.

Dataset	IVT [9]	ℓ_1 [10]	PN [5]	VTD [8]	MIL [4]	FRAG [7]	ASLSA [1]	Ours
Faceocc2	10.2	11.1	18.6	10.6	14.1	15.5	3.8	**4.0**
Caviar	66.2	65.9	53.0	60.9	83.9	94.2	2.3	**2.1**
Woman	167.2	131.6	9.0	136.6	122.4	113.6	2.8	**2.6**
Car11	2.1	33.3	25.1	27.1	43.5	63.9	2.0	**1.3**
David	3.6	7.6	9.7	13.6	16.1	76.7	3.6	**3.7**
Singer	8.5	4.6	32.7	4.1	15.2	22.0	4.8	**4.7**
Board	165.4	177.0	97.3	96.1	60.1	31.9	7.3	**7.8**
Average	60.5	61.6	35.1	49.9	50.7	59.7	3.8	**3.7**

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019

the center position of the ground truth. The CPE of each tracker on the six datasets is shown in Table 1. We can see the average center error of the eight trackers on seven datasets. Our results are marked in bold font. The average error of all the datasets is shown in the last row of the table. We can see that our results are almost the best on all the datasets.

The center location error only checks the deviation of the center point, which cannot detect the variation of pose and scale. Thus, we also exploit Pascal VOC overlap ratio as the second metrics to evaluate our results. The overlap ratio is defined as $R_{overlap} = (S_R \cap S_{GT})/(S_R \cup S_{GT})$, where S_R is the result bounding box area and S_{GT} is the ground truth bounding box area. Generally, it is considered to be a successful tracker if the VOC overlap ratio is greater than 0.5. The VOC overlap ratio of the trackers is shown in Table 2 and our results are marked in bold font. The average results are also given. We can see that our results are the best and all the VOC overlap ratios are larger than 0.8. In Fig. 5, we show the overlap ratios of each frame,

Table 2
The overlap ratio comparing with other seven trackers on seven sequences.

Dataset	IVT [9]	ℓ_1 [10]	PN [5]	VTD [8]	MIL [4]	FRAG [7]	ASLSA [1]	Ours
Faceocc2	0.59	0.67	0.49	0.59	0.61	0.60	0.82	**0.83**
Caviar	0.21	.020	0.21	0.19	0.19	0.19	0.84	**0.85**
Woman	0.19	0.18	0.60	0.15	0.16	0.20	0.78	**0.84**
Car11	0.81	0.44	0.38	0.43	0.17	0.09	0.81	**0.81**
David	0.72	0.63	0.60	0.53	0.45	0.19	0.79	**0.80**
Singer	0.66	0.70	0.41	0.79	0.33	0.34	0.81	**0.80**
Board	0.17	0.15	0.31	0.36	0.51	0.73	0.74	**0.83**
Average	0.48	0.43	0.43	0.43	0.34	0.33	0.80	**0.82**

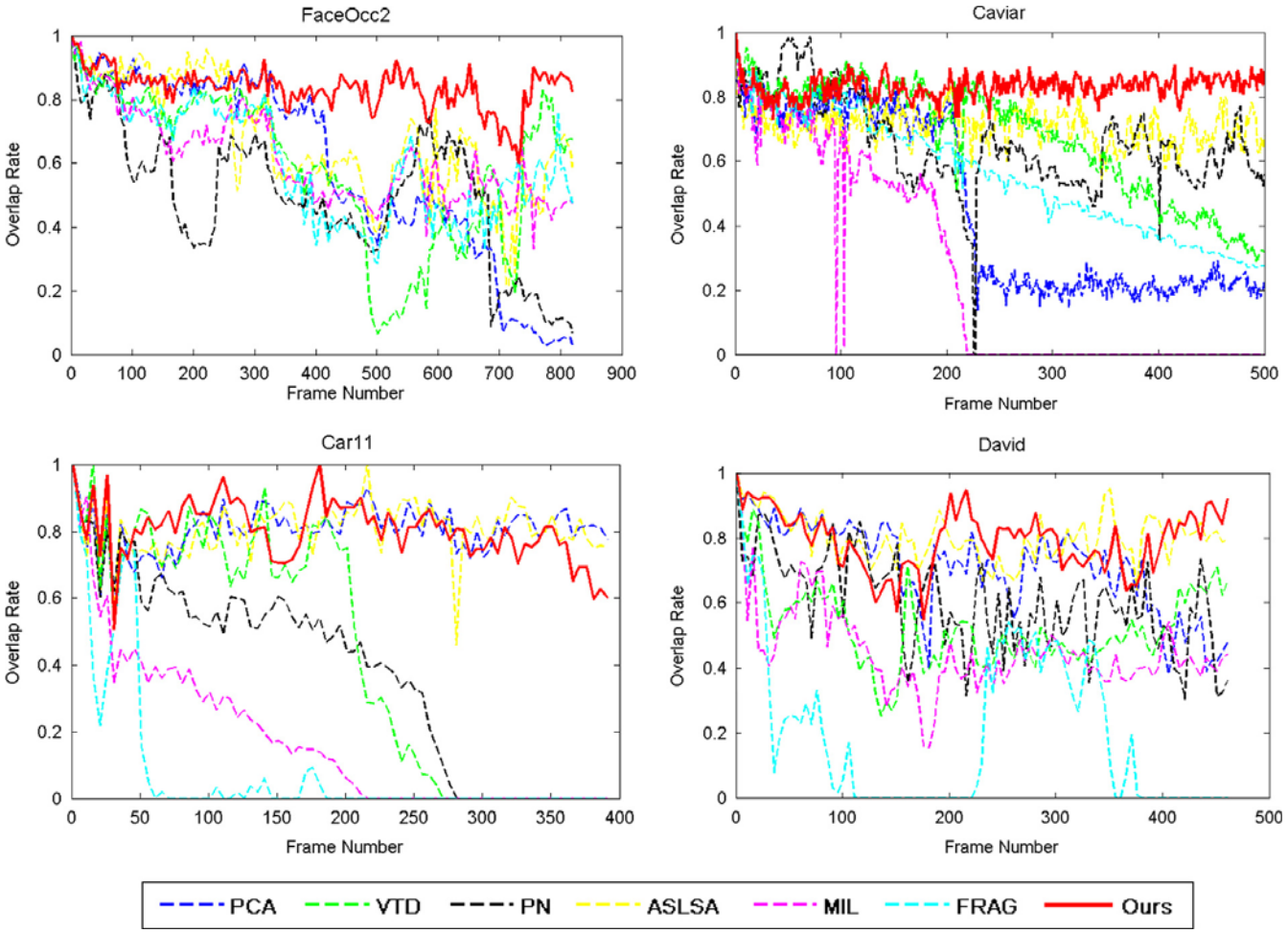

Fig. 5. Results for overlap rate of our tracker and six state-of-the-art trackers on four datasets. The solid line shows the results of our tracker. It is evident that our track obtains the best results and performs stably.

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019

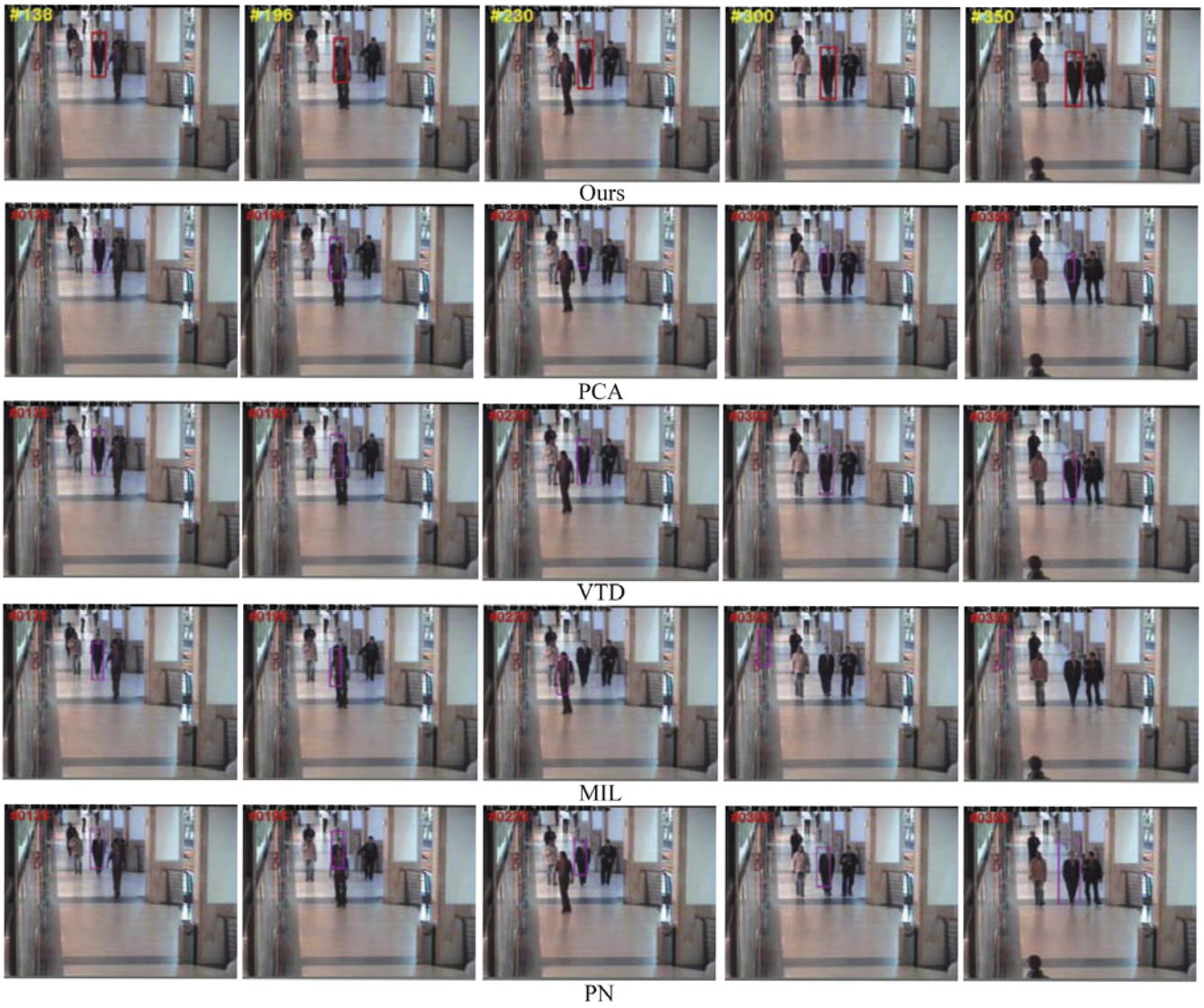

Fig. 6. Results comparison of our tracker and the four state-of-the-art trackers on dataset Caviar. The first line is our result, which performs the best.

Table 3
The average comparison of ASLSA and our tracker on four terms. Our results are in bold font.

Dataset	**speed** (**fps**)	Speed	**CPE**	CPE	**OR**	OR	**Drifting**	Drifting
Caviar3	**2.57**	2.60	**2.61**	33.1	**0.83**	0.48	**1/10**	6/10
Woman	**2.76**	2.86	**2.99**	63.2	**0.81**	0.56	**0/10**	4/10
Stone	**2.20**	2.25	**8.40**	14.4	**0.78**	0.76	**0/10**	2/10
Board	**3.27**	3.36	**3.23**	4.25	**0.57**	0.53	**0/10**	2/10
Average	**2.70**	2.77	**4.31**	28.7	**0.75**	0.58	**1/40**	14/40

which includes the six state-of-the-art trackers on four datasets. It is clear to see that our tracker performs the best and maintains stability along the sequences. The bounding box areas of five trackers in the original images are shown in different rows in Fig. 6, where comparison results on Caviar are shown. These datasets focus on variation of background and scale. We select the frames with occlusions and drastic changes. Our results are marked by red bounding boxes. We can see that the proposed tracker performs well in the frames with the occlusion.

Comparison and drifting: In the above results, all the trackers select the best results as their final results. The results with drifting are discarded. Therefore, the above results have not shown the robustness and the general performance of the trackers. In order to evaluate the trackers more persuasively, we continuously run the ASLSA tracker, which is a representative local sparse tracker, and our tracker 10 times. Then we give the comparison of average running time, frequency of drifting, average OR and average CPE in Table 3. The unit of the

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019

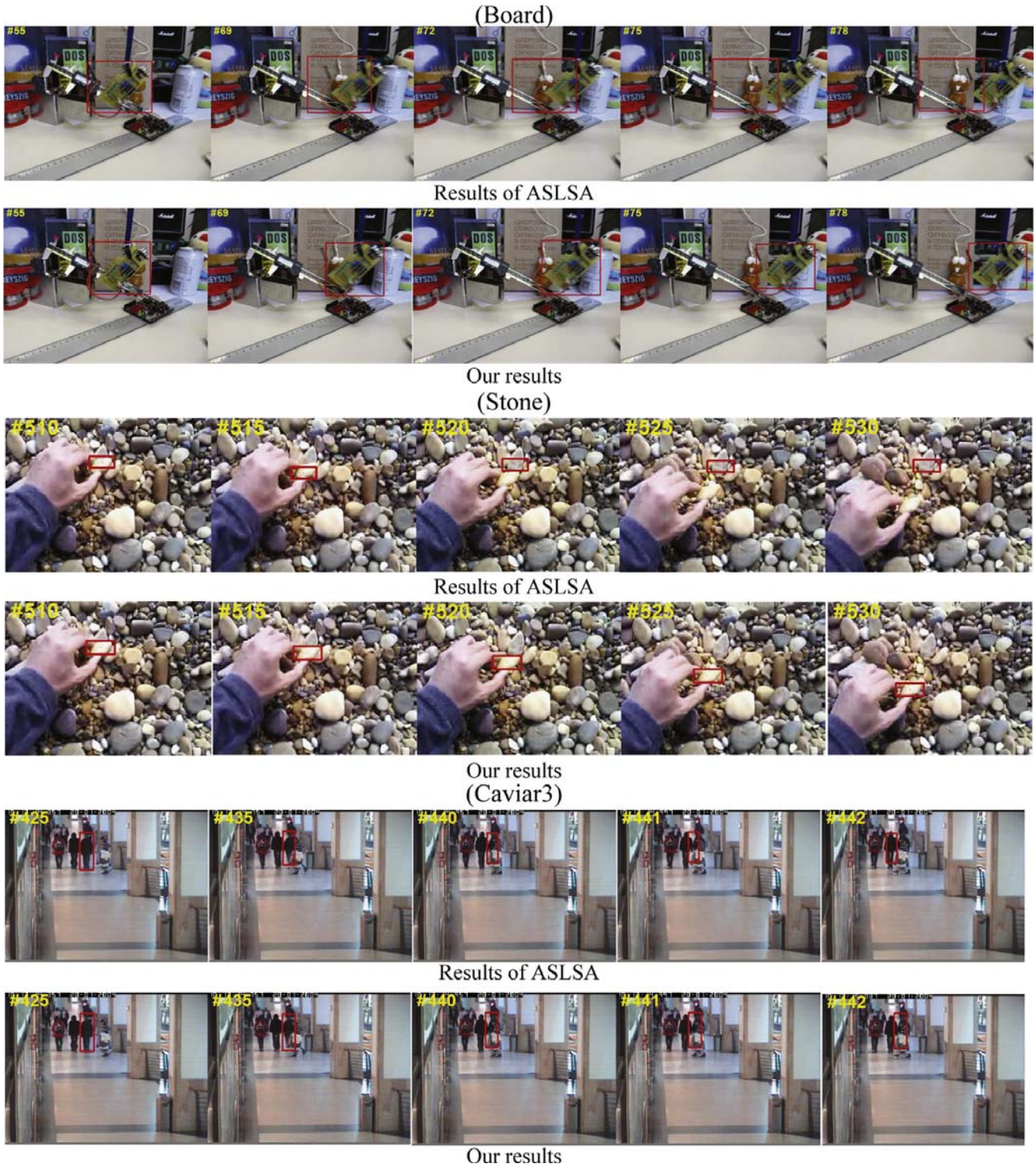

Fig. 7. Results comparison of ASLSA tracker and our tracker on datasets Board, Stone and Caviar3. These datasets contain occlusion and complicated background, which are the causes of drifting. We run the two trackers many times and show the comparison results in this figure. For each dataset, the first row is the result of ASLSA and the second row is our result. As we can see, the ASLSA tracker drifts away several times but our tracker obtains stable performance without drifting.

speed is frame per second (fps). We can see that the speed of our tacker is close to the speed of ASLSA. In the Drifting column, 1/10 indicates that the tracker drifts away one time in 10 tests. We can see that the ASLSA tracker drifts away frequently, but our tracker almost does not drift away. That is because we add a global constraint on the local model, which can alleviate the drifting problem effectively. The average results of our tracker have a big improvement compared to the results of ASLSA, no mater in terms of CPE or OR. The comparison results on original images can be seen in Fig. 7. We can see that the ASLSA tracker drifts away when occlusion or complicated

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019

background occurred. However, our tacker obtains stable performance in these cases. The comparison shows the significance of the global constraint model.

7. Conclusion

In this paper, we propose a novel model based on local sparse and global constraint. The proposed model improves the local sparse method which often occurs in overfitting and drifting problem, by adding a constraint on the distribution of the local contributions. The constraint is achieved by give a penalty to the candidate with larger variance of the distribution. And a two-step algorithm is given to implement the proposed model. In the template update stage, occlusion information is fully used in both selecting the template to be discarded procedure and reconstructing the observation procedure. Experimental results and comparisons with other state-of-the-art trackers on popular benchmark datasets show the superiority of our method in dealing with the drifting problem.

Acknowledgments

This study was supported by the Key Foundation Research Project of Shenzhen (No. JC201104210033A), Innovation Project of Scholars from Overseas of Shenzhen (No. KQCX20120801104656658), Technology Innovation Project of Shenzhen (No. CXZZ20120618155717337 and CXZZ20130318162826126), Shenzhen Strategic Emerging Industries Program (No. JCYJ20120613150552967), the Education Department of Hubei Province Science and Technology Research Project (No. B2014234), and NSFC (No. 61370213). The authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

References

[1] X. Jia, H. Lu, M.-H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: Conference on Computer Vision and Pattern Recognition, 2012, pp. 1822–1829.
[2] S. Avidan, Ensemble tracking, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 261–271.
[3] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: European Conference on Computer Vision, 2008, pp. 234–247.
[4] B. Babenko, M.-H. Yang, S. Belongie, Visual tracking with online multiple instance learning, in: Conference on Computer Vision and Pattern Recognition, 2009, pp. 983–990.
[5] Z. Kalal, J. Matas, K. Mikolajczyk, P-n learning: bootstrapping binary classifiers by structural constraints, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 49–56.
[6] S. Wang, H. Lu, F. Yang, M.-H. Yang, Superpixel tracking, in: International Conference of Computer Vision, 2011, pp. 1323–1330.
[7] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: Conference on Computer Vision and Pattern Recognition, 2006, pp. 798–805.
[8] J. Kwon, K.M. Lee, Visual tracking decomposition, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 1269–1276.
[9] D.A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (2008) 125–141.
[10] X. Mei, H. Ling, Robust visual tracking using l1 minimization, in: International Conference on Computer Vision, 2009, pp. 1436–1443.
[11] X. Mei, H. Ling, Y. Wu, E. Blasch, Minimum error bounded efficient l1 tracker with occlusion detection, in: Conference on Computer Vision and Pattern Recognition, 2011, pp. 1257–1264.
[12] B. Liu, J. Huang, L. Yang, C. Kulikowsk, Robust tracking using local sparse appearance model and k-selection, in: Conference on Computer Vision and Pattern Recognition, 2011, pp. 1313–1320.
[13] C. Hong, N. Li, M. Song, J. Bu, C. Chen, A level-set based tracking approach for surveillance video with fusion and occlusion, in: Image and Video Technology, 2010, pp. 156–161.
[14] C. Hong, J. Zhu, M. Song, Y. Wang, Realtime object matching with robust dominant orientation templates, in: International Conference on Pattern Recognition, 2012, pp. 1152–1155.
[15] J. Yu, D. Tao, M. Wang, Adaptive hypergraph learning and its application in image classification, IEEE Trans. Image Process. 21 (7) (2012) 3262–3272.
[16] J. Yu, Y. Rui, Y. Tang, D. Tao, High-order distance-based multiview stochastic learning in image classification, IEEE Transactions on Cybernetics PP 99 (2014). 1–1.
[17] D. Tao, X. Li, X. Wu, S. Maybank, Geometric mean for subspace selection, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 260–274.
[18] D. Tao, X. Li, X. Wu, S. Maybank, General tensor discriminant analysis and Gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1700–1715.
[19] J. Yu, Y. Rui, D. Tao, Click Prediction for Web Image Reranking Using Multimodal Sparse coding, 2014, pp. 2019–2032.
[20] J. Yu, R. Hong, M. Wang, J. You, Image clustering based on sparse patch alignment framework, Pattern Recognition 47 (11) (2014) 3512–3519.
[21] Z. Xiao, H. Lu, D. Wang, Object tracking with l2-rls, in: International Conference on Pattern Recognition, 2012, pp. 1351–1354.
[22] B.D. Lucas, T. Kanade, et al., An iterative image registration technique with an application to stereo vision, in: International Joint Conference on Artificial Intelligence, vol. 81, 1981, pp. 674–679.
[23] L. Matthews, T. Ishikawa, S. Baker, The template update problem, IEEE Trans. Pattern Anal. Mach. Intell. 26 (6) (2004) 810–815.
[24] A. Doucet, N. de_Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer, New York, 2001.
[25] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, J. Mach. Learn. Res. 11 (2010) 19–60.
[26] J. Santner, C. Leistner, A. Saffari, T. Pock, H. Bischof, Prost: parallel robust online simple tracking, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 723–730.

Please cite this article as: X. You, et al., A robust local sparse tracker with global consistency constraint, Signal Processing (2014), http://dx.doi.org/10.1016/j.sigpro.2014.09.019