International Journal of Hybrid Information Technology
Vol. 10, No.1 (2017), pp. 125-138
http://dx.doi.org/10.14257/ijhit.2017.10.1.11

Robust Visual Tracking via Collaborative Voting with Structured Sparse Representation

Yang Liu, Yibo Li, Xiaofei Ji and Yangyang Wang

College of Automation, Shenyang Aerospace University, Shenyang, China
yang97_net@sau.edu.cn

Abstract

Sparse representation based methods have recently attracted much attention in visual tracking due to the robustness to corruption, occlusion and other challenging issues. The target templates and the candidates have been applied as training samples and inputs for the method. However, the sparse coefficients can not be made full use to discriminate between the target and the background, hence tracking method may fail when there is similar object or occlusion in the scene. In this paper, we propose a novel visual tracking method based on collaborative voting with sparse representation. Different from previous methods, visual tracking is formulated as an object recognition problem in the proposed method, which makes the tracking task more robust to occlusion. The collaborative voting exploits the contributions of the bases in dictionary based on alignment-sum and alignment-max pooling operating, which take the special layout of local patches into consideration and improve the robustness of the proposed method. In addition, a weight vector is generated based on alignment pooling to measure the importance of a template. Then incremental subspace learning is used to learn a new template to replace the template with the lowest weight. Furthermore, the update scheme considers both the latest observation and the original template, thereby enabling the tracker to deal with appearance change effectively and alleviate the drift problem. The proposed tracker is empirically compared with state-of-the-art trackers on some challenging sequences. Both quantitative and qualitative comparisons show that the proposed tracker is effective and robust.

Keywords*: visual tracking, collaborative voting, structured sparse representation, object recognition, template update*

1. Introduction

Visual tracking is widely used in computer vision, such as video surveillance, human-computer interaction, robotic navigation, image compression and so on. Although much progress has been made in recent years, it remains a challenging problem due to large appearance change caused by factors such as partial occlusion, illumination change, pose change, background clutter and viewpoint variation.

Over the years, many tracking algorithms have been proposed to deal with these difficulties. Incremental visual tracking [1] is proposed to learn the dynamic appearance of the tracked target via incremental principal component analysis. Grabber and Bischof [2] present a tracking method using the online AdaBoost algorithm, which achieves real-time performance. Babenko,Yang and Belongie[3] put all ambiguous positive and negative samples into bags based on multiple instance learning(MIL) to learn a discriminative model for tracking. Among the challenging problems, serious occlusion is one of the most challenging (See Figure.1). A truly robust tracking method must be able to handle occlusion. However, modeling occlusion is not straightforward and non-trivial by far. Motivated by the success of sparse representation in face recognition[4], sparse representation has been successfully applied to visual tracking under the particle filter

ISSN: 1738-9968 IJHIT

framework[5] as an attempt to alleviate the occlusion problem and improves the performance of visual tracking. In [6, 7] a target candidate is sparsely represented by a linear combination of the atoms of a dictionary composed by dynamic target templates and trivial templates. By introducing trivial templates, the tracker can handle partial occlusion. The sparse representation problem can be solved through l_1 minimization with non-negativity constraints. Up to now, many improved tracking algorithms have been proposed. We consider these algorithms based on sparse representation lying in two categories as generative or discriminative methods. The generative methods proposed in [8-12] use appearance models and formulate the problem as finding the image observation with minimal reconstruction error. Discriminative methods [13-15] regard the target tracking as a binary classification problem and distinguish the target from the background. The trained classifier is used to discriminate the target from background. Several algorithms [16, 17] that exploit the advantages of both generative and discriminative models have been proposed. The method [18] solved in closed-form has been proposed based on non-sparse linear representations. A new minimization model for sparse representation is proposed to improve the accuracy and efficiency [19].The relationship between particles is exploited and matrix learning is proposed for tracking in [20]. In this paper, we focus on developing a robust algorithm using a generative appearance model that takes occlusion and appearance changes into consideration to alleviate tracking drift.

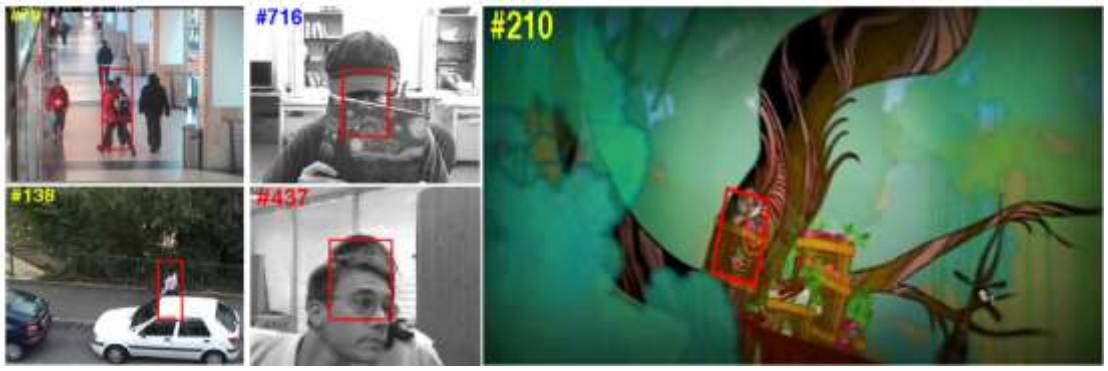

Figure 1. Challenging Videos Including Serious Occlusions

Different from previous algorithms, we formulate the tracking task as a recognition problem. It means that the roles of templates and candidates are reversed in l_1 minimization. In the previous work, visual tracking is considered as a searching problem which checks all the candidates one by one. All the candidates are represented by a linear combination of the templates. The combination coefficients are obtained by using the minimum l_1-norm solution. The result is the candidate that has the most similar appearance with the target template. The methods are complex and time-consuming. Motivated by the successful application of sparse representation in face recognition, we consider tracking task as recognition problem, in which templates and candidates are used as observed samples and dictionary. The proposed framework can deal with occlusion and corruption uniformly by exploiting the fact that the errors due to occlusion and corruption are often sparse with respect to the standard basis.

Motivated by the success of sparse representation for image classification as well as visual tracking [21], we present a robust tracking method using a collaborative voting based on a structured local sparse coding model that takes the spatial information of patches into account. Firstly, the proposed method samples overlapped local image patches within the target region. Secondly, we obtain the response matrix of patches in templates to all bases in dictionary. Next, the collaborative voting consists of two voting

processes. One voting process is to measure the contribution of each basis in dictionary to all the observed patches and cast votes to candidate with more contribution, which is based on alignment-sum pooling operating. The other voting process is to measure the contribution of alignment bases to a specific observed patch and cast votes to candidate with more contribution, which is based on alignment-max pooling operating. Finally, the collaborative voting is used as the probability of a candidate as tracking result.

The appearance of a template often changes significantly during the tracking process. Therefore, the update scheme is important and necessary. Since our representation is constructed at the pixel level, misalignment between templates and candidates might lead to degraded performance. To alleviate this problem, one of two strategies can be employed [22]: (1) Templates can be constructed by an over completely dictionary of the target object, which includes transformed versions of both. (2) Columns of candidates can be aligned to columns of templates as in [23]. In this paper, we employ the second strategy. The templates are updated based on both incremental subspace learning and sparse representation. The weights of the templates are introduced and the template with the lowest weight is replaced, which reduces the influence of the template with partial occlusion. The update scheme facilitates the tracker to account for appearance changes and makes the proposed method more accurate and robust.

2. Collaborative Voting Based on Structured Sparse Representation

In this section, the details of the proposed method are presented. Firstly, we formulate visual tracking as a recognition problem. The main point is that we reverse the roles of the templates and all the candidates in sparse representation. The framework can handle errors due to occlusion and corruption uniformly by exploiting the fact that these errors are often sparse with respect to the standard basis. Secondly, the collaborative voting based on structured sparse representation is discussed in detail. Intuitively, a basis is more similar to template when its response is bigger, and we consider the basis with bigger contribution in sparse representation. Be driven by this assumption, we exploit the contribution of each basis in dictionary and cast votes to candidates with bigger contribution.

2.1. Problem Formulation

The tracking task is formulated as a recognition problem. It means that we consider a candidate as a sample of a class. All the candidates form the training samples, and we must classify a template or some templates into a candidate. Just as the method in [4], an observed template is represented by the linear combination of all the candidates. Given a template y, the target state x_{t-1} in the time instant $t-1$, the particle states $x_t^1, x_t^2, \ldots, x_t^N$ are randomly sampled around the state x_{t-1} according to a zero-mean Gaussian distribution in the current time t and $C=[C_1, C_2, \ldots, C_N]$ denotes target candidates. If we assume that the target templates and all target candidates lie in a special low-dimensional feature space, often called a manifold, then the target template can be represented by a linear combination of all target candidates, see Equation (1):

$$y = \alpha_1 C_1 + \alpha_2 C_2 + \cdots + \alpha_N C_N \tag{1}$$

where the elements of y and C_i are the feature values (for example the gray values), $\alpha=[\alpha_1, \alpha_2, \ldots, \alpha_N]$ denotes the coefficient vector whose entries are zero except one (corresponding to the jth candidate) having the most similar appearance with the target template in ideal condition. In fact, the coefficient vector is sparse due to better representing the observed template. Then, the linear representation of y can also be rewritten in matrix form as Equation (2):

$$y = C\alpha \tag{2}$$

Because the linear system (2) is usually underdetermined, its solution is not unique. Recent development in the emerging theory of sparse representation and compressed sensing reveals that if the sparse code, *i.e.* α, is sparse enough, the solution of Equation (2) is equal to the solution to the following l_1-minimization problem (3):

$$\arg\min_{\alpha} \|y - C\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{3}$$

where λ is a tradeoff parameter between reliable representation and joint sparsity regularization. If given some template images $T = [T_1, T_2, \ldots, T_n]$ as observed samples, then the linear representation of T can be rewritten in matrix form as Equation (4):

$$\arg\min_{\alpha_1,\alpha_2,\ldots,\alpha_n} \sum_{i=1}^{n} \|T_i - C\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1 \tag{4}$$

In general, taking local patches as feature makes the algorithm more robust than the complete image as feature in many researches such as classification, recognition and also tracking tasks. When sample M local patches inside all the template and candidate images, a local patch of a template is represented by the linear combination of all the local patches of the candidates. Then, the representation of T is computed by the Equation (5):

$$\arg\min_{\alpha_1^1,\alpha_1^2,\ldots,\alpha_n^M} \sum_{i=1}^{n}\sum_{j=1}^{M} \|y_i^j - D\alpha_i^j\|_2^2 + \lambda\|\alpha_i^j\|_1 \tag{5}$$

where $D = [d_1, d_2, \ldots, d_{N\times M}]$, d_i is a local patch sampled in candidates. y_i^j denotes the jth local patch of the ith template, and α_i^j denotes the corresponding coefficient vector. In the proposed method, we sample overlapped local image patches inside the templates and candidates. Then all the above operations are conducted. In the recognition framework, a local patch of a template is represented by the linear combination of training patches coming from candidates. Then we can obtain the coefficients of the local patch. Next, we cast a vote to the candidate which holds the biggest coefficient based on structured sparse representation. Finally the candidate with the most votes is recognized as the tracking result. The details are shown in subsection 2.2.

2.2. Collaborative Voting Based on Alignment-max and Alignment-sum Pooling

We propose collaborative voting for visual tracking. We measure the contribution of a basis in dictionary to all local patches in T based on alignment-sum pooling operating and cast votes according to contributions. In addition, we use the coefficients of a local patch in a certain position in T to all the patches with according spatial layout inside candidates to cast votes based on alignment-max pooling. Finally, we propose a collaborative voting within the particle filter framework.

Alignment-sum pooling for a basis to all observed patches

A row of response matrix shows the contribution of a basis to all observed samples in representation task. We cast a vote to some candidate based on this fact. Let $D = [d_1, d_2, \ldots, d_k] \in R^{m\times k}$, where $k = M \times N$,and $A = [a_1, a_2, \ldots, a_r] \in R^{k\times r}$, where a_i denotes the sparse vector of ith observed patch in T, $r = n \times M$. We want to check the contribution of a basis in dictionary. Since each row of A corresponds to the response of all local patches in T to one specific basis in dictionary D. There are some pooling methods may be used for image statistics. The sum pooling has shown good performance in image classification. In order to make the representation more robust, we use the sum pooling function on the sparse coefficients.

$$b_i = \sum_{j=1}^{r} a_{i,j} \tag{6}$$

Note b_i is the element of vector $b \in R^{k\times 1}$. We rewrite $b = [h_{1,1}, h_{1,2}, \ldots, h_{N,M}]^T$, where $h_{i,j}$ denotes the sum pooling operator of the response of the jth patch in the ith candidate. The contribution that the basis hold is bigger in representation task, the basis appearance is more similar to observed patch. Only sum pooling operating is not enough to improve the discriminative ability of sparse coefficients. Therefore, alignment-sum pooling is used to obtain better performance. We compare the contributions of bases at the fixed position and find the most one to cast vote.

$$\arg\max_{p} \left\{ h_{p,s} \right\}, p = 1, 2, \ldots, N \tag{7}$$

It means to find the maximal contribution for all the patches in all candidates. Then we cast a vote to the pth candidate. The alignment-sum pooling is well established with biophysical evidence in visual area and it is effective for image classification with local features. After checking all the bases in dictionary, the ith candidate holds some number of votes $conr_i$. The probability can be computed by Equation (8)**:**

$$P_r = \lambda_r . \frac{1}{exp(-conr_i)} \tag{8}$$

where λ_r is a constant defined by user.

Alignment-max pooling for an observed patches to all alignment basis

A column of response matrix shows the contribution of all bases to a specific observed sample in representation task. We vast a vote to some candidate based on this fact. Motivated by the success of sparse coding for object tracking[21], we propose our voting algorithm based on alignment-max pooling. The aim of this work is to check the contribution of all the alignment basis in dictionary to the corresponding observed patch in T. For an observed patch that is the sth patch sample inside target region, we can obtain its sparse vector $a_i = [a_{1,i}, a_{2,i}, \ldots, a_{k,i}]^T$, the ith column of A. Let $a_i = [v_{1,1}, v_{1,2}, \ldots, v_{1,M}, v_{2,1}, \ldots, v_{N,M}]$, where $v_{i,j}$ is the coefficient of the i th candidate and the j th patch. We use alignment-max pooling to find the biggest contribution of a basis in the corresponding position with the observed patch.

$$\arg\max_{q} \left\{ v_{q,s} \right\}, q = 1, 2, \ldots, N \tag{9}$$

It means that the basis of the qth candidate has the most similar appearance with the observed patch, and we cast a vote to the qth candidate. The voting process is involved in all the patches in T. Finally, each candidate holds some number of votes $conc_i$, which demonstrates the probability of a candidate as tracking result.

$$P_c = \lambda_c . \frac{1}{exp(-conc_i)} \tag{10}$$

where λ_c is a constant defined by user. The alignment-max pooling operating makes the representation more robust to occlusion and noise. The spatial structure of patches is taken into consideration. The spatial layout of local patched represent the structure of the target. Each local patch represents one fixed part of the target object. Without occlusion, a local path can be best described by the patches at the same positions of the candidates. When the candidates contain the target object with some appearance variation, the blocks that appear frequently in these candidates (as indicated by their sparse codes) should be

weighted more than others for more robust representation. By this voting operation, the spatial information of a single local patch is considered, which can decrease the negative influence brought by occlusion and improve the tracking performance. Figure.2 shows an example of the coefficients of a local patch with occlusion and a local patch without occlusion. In the proposed method, we only hold the coefficients obtained from the training patches at the same position, which improves the tracking performance due to the consideration of spatial layout of local patches.

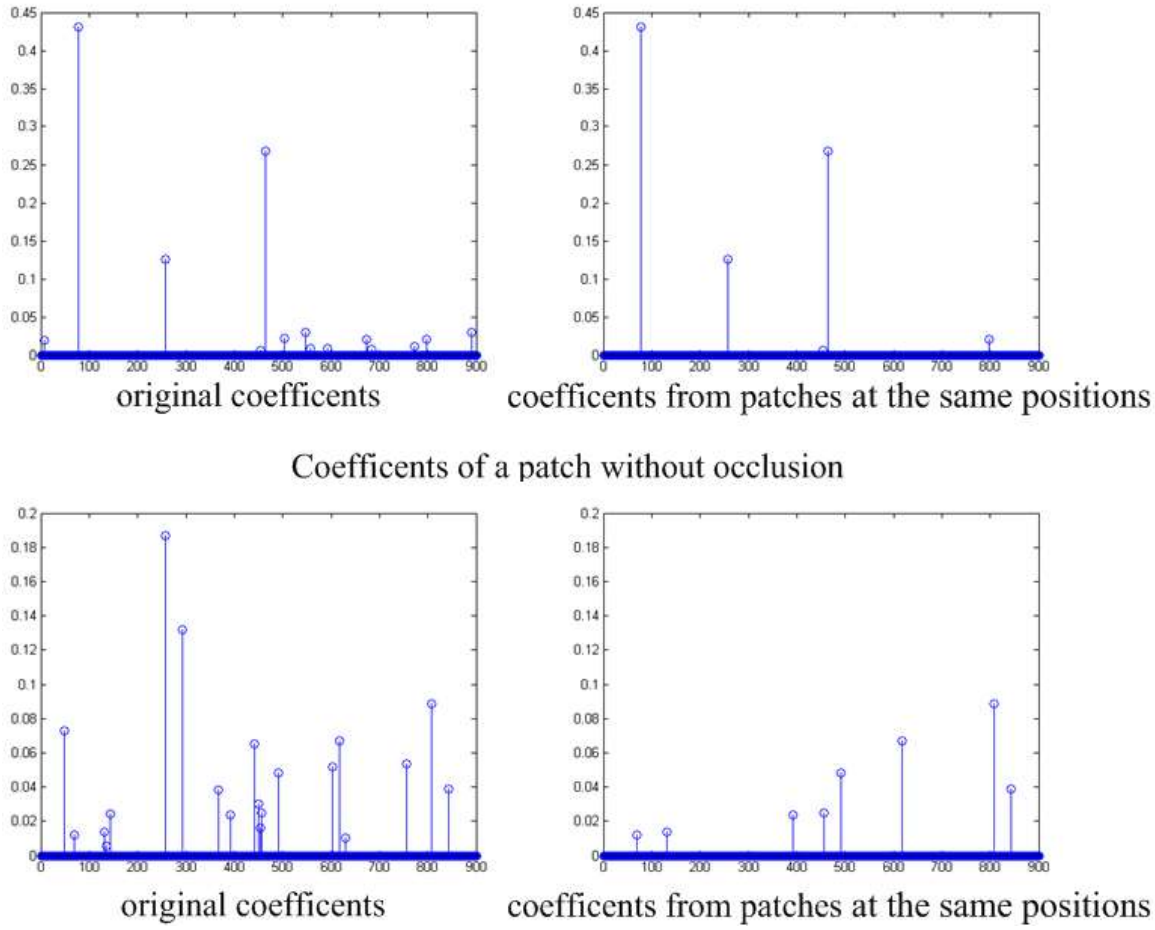

Figure 2. Coefficients of a Patch

If the number of the patches of templates is too small, it is difficult to track target successfully due to the fact that the votes are too dispersed to obtain the candidate with the most votes. It may occur that two candidates have the same number of votes, in addition, the number of votes is small. To address this problem, we take ten templates, *i.e.* $n = 10$, and sample nine patches in a template, *i.e.* $M = 9$. Then we have 90 times to cast a vote, which is enough to recognize the target from the candidates.

3. Template Update

If we do not update the template, a fixed template is not sufficient to capture appearance variations due to occlusion or pose changes in the video. Therefore, it is necessary and important to update the templates. While a rapidly changing model is susceptible to drift. Small errors are introduced each time when the template is updated. The errors are accumulated and the tracker drifts away from the target. We tackle this

problem by dynamically updating the target template set T. In this paper, subspace learning into sparse representation and alignment-pooling are introduced to adapt templates to the appearance change of the target, and reduce the influence of the occluded target template as well.

The larger coefficient of α is, the bigger contributions of the corresponding candidate is needed in the approximation $\|T - C\alpha\|_2$. We exploit the characteristic by introducing a weight w_i associated with each template T_i. Intuitively, the larger the weight is, the more important the template is. Set the weight vector $W = [\mathrm{w}_1, w_2, \ldots, w_n]^T$ for the templates. Note $\hat{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_{n]}$ represents the coefficients of templates corresponding to a candidate, where α_i is the coefficients of the ith template. Each template and candidate has several local patches, for example 9, thus α_i is a 9×9 matrix. Each local patch at a certain position of a template is represented by patches at different positions of all the candidates. The local appearance variation of a patch can be best described by the blocks at the same positions of the candidates (*i.e.* using the coefficients with the aligned positions). Based on this alignment-pooling[12], compute w_i using Equation (11)：

$$w_i = \sum(\mathrm{diag}(\alpha_i)) \tag{11}$$

We update the templates every m frames (5 in our experiments) from the current tracking result. The weights w_i of the m frames are added together to be final weights for a template, as shown in Equation (12). The template corresponding to the minimum of w_i is selected to be deleted and the target template set T is then updated using new template.

$$w_i = \sum_{f=1}^{m} \sum (\mathrm{diag}(\alpha_i)) \tag{12}$$

Sparse representation and subspace learning are combined to model the new template to update the target template set T. The tracking results of the target (5 in our experiments) are collected and thrown into the incremental learning method proposed in [2]. The result candidate can be modeled by a linear combination of the PCA basis vectors, which preserve the main and common information of the observed image. Trivial templates are introduced Motivated by the method proposed in [1].

$$y = D_t \omega + e = [D_t \quad I_d] \begin{bmatrix} \omega \\ e \end{bmatrix} \tag{13}$$

In Equation (13), where y denotes the observation vector, D_t is the matrix composed of eigenbasis vectors, ω is the coefficients of eigenbasis vectors and e indicates the pixels in that are corrupted or occluded. The nonzero entries of e indicate the pixels in y that are corrupted or occluded. Because particles are densely sampled around the current target state, the representations of templates with respect to D_t will be sparse (few candidates are required to represent them) and similar to each other in general. Therefore, we need to solve the convex optimization problem as l_1 regularized least square problem in Equation(14)：

$$\min_q \|y - Bq\|_2^2 + \lambda \|q\|_1 \tag{14}$$

where λ is a tradeoff parameter between reliable reconstruction and joint sparsity regularization, $B = [D_t \quad I_d]$, $q = [\omega \quad e]^T$. The coefficients of trivial templates are

employed to account for noise or occlusion and avoid much occlusion to be updated into the template set. The reconstructed image is then used for updating the template to be replaced.

Figure 3 shows some examples of templates. When the target is not occluded, the templates can adapt to the appearance change of the target. When the target is occluded, the templates focus on the parts which are not contaminated. In addition, the small number of templates characterizes the occluded target. With the proposed template update strategy, the proposed method can adapt to the appearance change of the target and handle the serious occlusion as well. The template update strategy is summarized in Algorithm 1.

Algorithm 1 Template Update

1: y is the newly chosen tracking target.
2: Throw y into the incremental learning method to compute new eigenbasis vectors.
3: Solve (12) and obtain ω.
4: Compute w_i using (10) for templates.
5: Set $w_1 = \inf$, which means we keep the first template fixed.
6: Copmpute $p = \min_i w_i$.
7: Detelet the template p.
8: Add $\widehat{y} = D_t\omega$ to the end of the template set T.
9: Normalize the template set T.

(a) Templates When the target is not occluded (The template with red box is deleted, and the one with green box is the new template learned by the proposed template update strategy.)

(b) Templates When the target is occluded (The template with red box is deleted, and the one with green box is the new template learned by the proposed template update strategy.)

Figure. 3 The Examples of Templates

4. Visual Tracking based on Particle Filter

Particle filters have been used extensively in object tracking task[24]. In this paper, particle filters is employed to track the target object. Let x_t and y_t denote the state variable and its observation of an object at time t respectively. The sparse representation model aims at calculating the observation likelihood for sample state x_t, *i.e.* $p(y_t|x_t)$. In general, it should reflect the similarity of a particle and the object templates while being robust against occlusion or appearance changes. At the frame t, given the target template set $T=[y_t^1, y_t^2, \ldots y_t^n]$, let $S_t=[x_t^1, x_t^2, \ldots, x_t^n]$ denote the sampled states and $D_t=[d_t^1, d_t^2, \ldots, d_t^k]$ denote the corresponding candidate target patch in target template space. Then, the optimal state x_t^* of frame t can be obtained by Equation(15)

$$x_t^* = \arg\max_{x_t^i \in S_t} p(y_t \mid x_t^i) \tag{15}$$

We apply the affine transformation with six parameters to model the target motion between two consecutive frames. The state transition is formulated as $p(x_t \mid x_{t-1}) = N(x_t; x_{t-1}, \Sigma)$, where Σ is a diagonal covariance matrix whose elements are the variances of the affine parameters. In the proposed method, the observation model is constructed by Equation(16)

$$p(y_t \mid x_t^i) = P_r \cdot P_c \tag{16}$$

where the right side of the Equation denotes the similarity between the candidate and the target based on the vote value. With the template updated incrementally, the observation model is able to adapt to the appearance change of the target.

5. Experimental Results

In this section, we compare the visual tracking performance of the proposed tracker with several state-of-the-art trackers on eight challenging video sequences, in which the challenges include serious occlusion, background clutter, illumination variation, pose variation, and scale change. The proposed approach is compared with eight state-of-the-art tracking methods including incremental visual tracking (IVT) method[1], L1 tracker[6], structural local sparse appearance model(LSAM)[21], multiple instance learning (MIL) tracker[3], online multiple support instance tracking (OMSIT) method[25], compressive tracking(CT)[14], learning a deep compact image representation(DLT)[26] and object tracking with local sparse representation (OLSR)[13]. We downloaded the source codes of these methods from the websites of their authors. The proposed algorithm is implemented in MATLAB and runs at 0.1 seconds per frame compared to 0.2 seconds per frame on a Pentium 3.2 GHz Dual Core PC with 3.4 GB memory. The l_1 minimization problem is solved with the SPAMS package [27].

5.1. Parameter Setting

The numbers of object templates and the particles are set as 10 and 200, respectively. In many other methods, the number of particles is 600. However, more particles is not suitable in the proposed method due to the fact that more candidates make the errors between different classes is too small to recognize the accurate result. We set the template size to 32×32 and extract overlapped 16×16 local patches within the target region with 8 pixels as step length. The regularization constant is set to 0.001 in all experiments. As for the templates update, 8 eigenvectors are used to carry out incremental subspace learning method in all experiments. And we update templates every 5 frames.

5.2. Quantitative Evaluation

To quantitatively assess the performance of the proposed trackers, we use two common performance metrics for quantitative comparison: relative overlap area and central pixel error. The center location error is defined as the Euclidean distance (in pixels) between the central location of the tracked target and the manually labeled ground truth data. The relative overlap area [28] is defined by the Equation (20) to evaluate the success rate.

$$rate = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{20}$$

where B_p and B_{gt} denote the predicted bounding boxes and the ground truth boxes respectively. The value of the rate is bigger, tracking is considered more successful. The quantitative comparison results are summarized in Table 1 and Table 2.

Table 1. The Comparison of Average Center Error (in pixels)

	panda	animal	face	woman	girl	singer	car	caviar
CT	107.7	231.6	16.8	107.6	35.5	18.7	27.3	64.3
L1	120.7	189.8	18.7	130.3	12.8	3.8	33.5	25.5
LSAM	15.4	6.1	3.9	2.5	12.1	4.0	2.2	5.7
MIL	89.1	262.4	14.9	137.2	30.0	23.8	46.4	65.9
OLSR	15.4	10.2	47.1	147.2	14.0	4.2	5.6	57.4
DLT	14.2	9.4	23.2	4.6	11.9	2.9	1.7	55.6
IVT	143.6	7.0	22.1	172.5	66.1	10.5	2.2	70.2
OMSIT	142.8	115.2	38.0	147.3	143.1	9.1	67.3	73.8
Ours	2.9	6.7	4.2	3.0	12.5	3.8	2.2	5.5

Table 2. The Comparison of Success Rate

	panda	animal	face	woman	girl	singer	car	caviar
CT	15.8	3.0	59.7	17.0	48.0	34.0	35.3	13.2
L1	3.4	4.8	58.6	16.3	65.2	78.9	44.3	4.3
LSAM	75.2	62.9	82.3	79.0	68.4	82.9	80.9	73.2
MIL	24.7	5.9	58.4	14.9	54.1	30.2	2.6	13.5
OLSR	74.1	60.0	52.9	17.2	67.7	78.8	52.0	13.3
DLT	67.0	60.6	50.5	70.1	66.6	83.2	74.2	15.7
IVT	45.4	62.0	42.7	18.4	14.0	53.2	80.6	14.1
OMSIT	19.4	20.1	40.8	5.9	12.2	56.7	9.4	11.9
Ours	79.8	62.8	83.8	83.3	64.5	83.1	81.2	75.4

Figure 4 shows the relative position errors (in pixels) between the center and the tracking results. These results indicate that the overall performance of the proposed tracker outperforms the most of other trackers in these sequences, especially for the videos with serious occlusion.

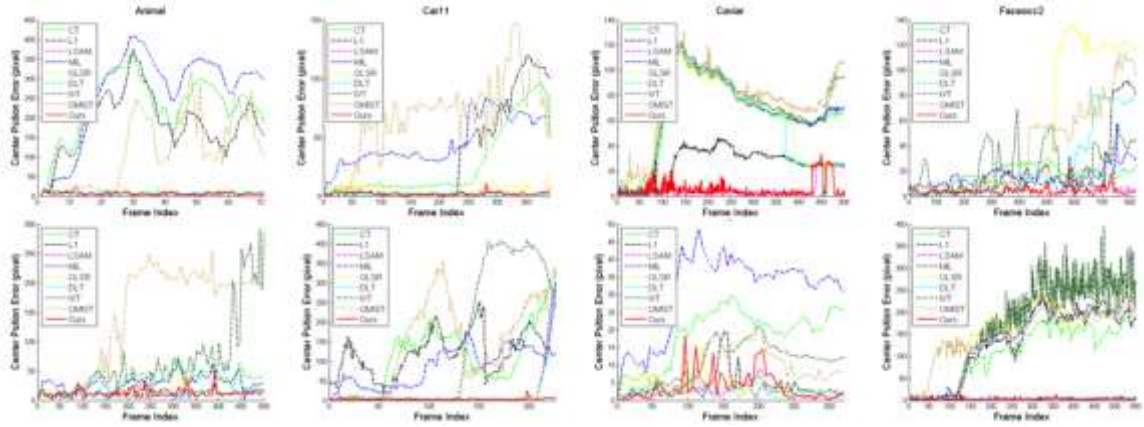

Figure 4. Quantitative Evaluation in Terms of Center Location Error (in pixel)

5.3 Qualitative Evaluation

Figure 5 shows csome key frames with bounding boxes reported by all eight trackers for each of the 8 video sequences.

In the animal sequence, the target is a fast moving animal with motion blur. LSAM, OLSR, DLT, IVT and the proposed tracker can track the target to the end. The other four trackers fail when some similar objects appear in the background.

In the car11 sequence, the tracked target is a car moving on a road where the environment is very dark. Since the similarity between the car and background, L1, MIL, OMSIT drift away from about the frame 50 and CT can not track the car from about the frame 260. LSAM, DLT, IVT and the proposed tracker can also track the car accurately.

In the Caviar sequence, the tracked target is a man walking in a piazza. The man is severely occluded three times by the other persons. The numerous methods fail to track the target because there are similar objects around it when heavy occlusion occurs. When appear occlusion first time, the methods except LASM, L1 and the proposed tracker track the person with red dress. When appear occlusion second time, L1 tracks the person beside the target from frame 118. Only LSAM and our tracker do not drift away from beginning to frame 439 when appear occlusion again. From 439 to 477, LSAM and our tracker drift away and track the person beside the target. And from frame 477 to end, our tracker tracks the target successfully due to the fact that the structured local information and the robust template update scheme can distinguish the target and similar objects.

In the faceocc2 sequences, the challenging are the serous occlusion and in-plane rotation. When the face is occluded by the book from the side, IVT fails at frame 284 because the occlusion. OMSIT and OLSR fail at the frame 433 and the frame 516 because of the occlusion with in-plane rotation. The tracking box of IVT, L1 and DLT are gradually narrowing, which leads to failure tracking. LSAM, CT, MIL and our trackers can track the face accurately along the whole sequence.

In the girl sequences, the challenging consists of occlusion, in-plane and out-of-plane rotations. The IVT and OMSIT trackers can not deal with out-of-plane rotation. The CT tracker drifts away when the girl is occluded. The LSAM, L1, MIL OLSR, and DLT can track the girl accurately. Our tracker hold plain performance when deal with in-plane and out-of-plane rotations, which can be concluded from the quantitative evaluation.

In the panda sequence, the target experiences more and larger in-plane rotations in addition to occlusion. CT, L1, MIL and OMSIT tracker drift after the target undergoes large rotations whereas our method performs well throughout this sequence. IVT tracker fails to detect occlusions and tracks the target object after frame 132. Furthermore, LSAM, DLT and OLSR drift away when the sequence undergoes occlusions again at frame 213. While our tracker still performs well owing to the historical and the updating templates.

In the singer1 sequence, the stage light changes drastically. All trackers can track the target but CT and MIL do not support scale change and hence the results are less satisfactory.

In the woman sequence, the tracked target is a woman walking in the street. The woman is severely occluded several times by the parked cars. OMSIT first fails at frame 45 because of the pose change and the background change. All other trackers except DLT and our trackers fail when the woman walks close to the car at about frame 130. DLT can follow the target accurately but with reduced tracking box. LSAM and our tracker perform well accurately along the entire sequence.

In summary, our tracker can deal with serious occlusion, illumination change, motor blur, rotation and background clutter. Especially when the targets undergo serious occlusion (woman, caviar, panda), our tracker demonstrates satisfactory performance.

6. Conclusion

In this paper, we propose a novel visual tracking method based on structured local information and collaborative voting scheme with the considering visual tracking as recognition problem. Different from the traditional methods, the set of candidates is used as training samples and the goal is to recognize the target template from all target candidates. The collaborative voting exploits the contribution of each basis in dictionary and takes the spatial layout of local patches into consideration, which helps distinguish

target. The candidate with the more votes has higher probability as the tracking result. Besides, sparse representation is combined with incremental subspace learning and alignment-pooling for template update, which reduces drifts and enhances the proposed method to adaptively account for appearance change in dynamic scenes. Experiments have been conducted on some challenging benchmark video sequences. The quantitative and qualitative comparisons with several state-of-the-art methods demonstrate the effectiveness and robustness of the proposed algorithm. However, one drawback in the proposed method is the stability for that we use only 200 particles to track target. In the future, the velocity, acceleration and orientation of the target associated with the time are considered in sowing particles to prevent drifting problem.

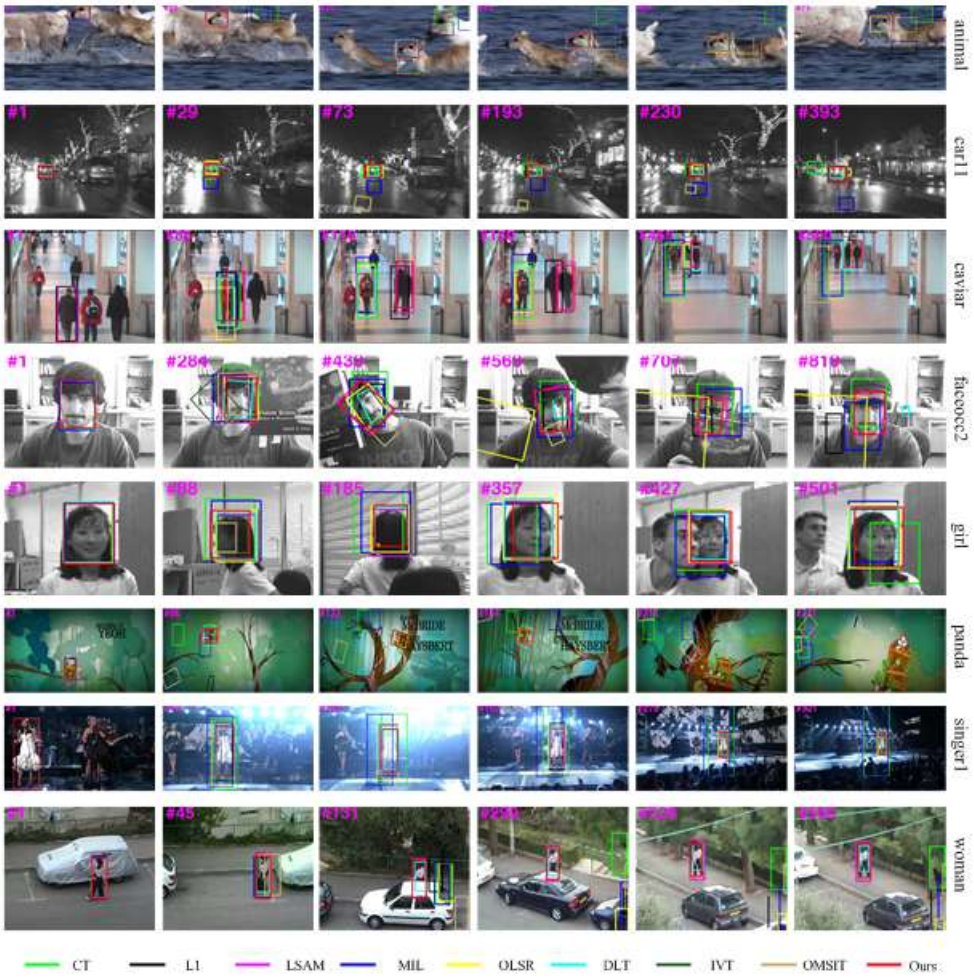

Figure. 5 Qualitative Evaluations in Terms of the Bounding Box

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant, China (No.61103123), the Scientific Research General Project of Education Department of Liaoning Province, China (No. L2014066) and Provincial Natural Science Foundation of Liaoning Province with the Grant (No. 2015020101).

References

[1] D. Ross, J. Lim, R.-S. Lin and M.-H. Yang, "Incremental learning for robust visual tracking", International Journal of Computer Vision, vol. 77, no. 1, **(2008)**, pp. 125-141.
[2] H. Grabner and H. Bischof, "On-line boosting and vision.In Proceedings of Conference on Computer Vision and Pattern Recognition", IEEE, New York, USA, **(2006)**, pp. 260-267.
[3] B. Babenko, M.-H. Yang and S. Belongie, "Visual tracking with online multiple instance learning", In Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, Miami, Florida, USA, **(2009)**, pp. 983-990.
[4] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, "Robust Face Recognition via Sparse Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 1, **(2009)**, pp. 210-227.
[5] A. Doucet, N. de Freitas and N. Gordon, "Sequential Monte Carlo Methods in Practice. Springer-Verlag", 2nd ed., J. Peters, Ed., New York, USA: McGraw-Hill, **(2001)**, pp. 15-64.
[6] X. Mei and H. Ling, "Robust visual tracking using L1 minimization", International Conference on Computer Vision,Kyoto, Japan, **(2009)**, pp. 1437-1443.
[7] X. Mei, H. Ling, Y. Wu, E. Blasch and L. Bai, "Minimum error bounded efficient L1 tracker with occlusion detection", In Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, **(2011)**, pp. 1257-1264.
[8] B. Liu, J. Huang, L. Yang and C. Kulikowski, "Robust visual tracking using local sparse appearance model and K-selection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 12, **(2013)**, pp. 2968-2981.
[9] X. Zhang, W.Li, W. Hu, H. Ling and S. Maybank, "Block covariance based L1 tracker with a subtle template dictionary", Pattern Recognition, vol. 46, no. 7, **(2013)**, pp. 1750-1761.
[10] D. Wang, H. Lu and M.-H. Yang, "Online object tracking with sparse prototypes", IEEE Transactions on Image Processing, vol. 22, no. 1, **(2013)**, pp. 314-325.
[11] T. Bai and Y.F. Li, "Robust visual tracking with structured sparse representation appearance model", *Pattern Recognition*, vol. 45, no. 6, **(2012)**, pp. 2390–2404.
[12] S. Zhang, H. Yao, X. Sun and S. Liu, "Robust visual tracking using an effective appearance model based on sparse coding", *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 3, **(2012)**, pp. 1–18.
[13] W. Qing, C. Feng and X. Wenli, "Online Discriminative Object Tracking with Local Sparse Representation", In Proceedings of the 2012 IEEE Workshop on the Applications of Computer Vision, **(2012)**, pp. 425-432.
[14] K. Zhang, L. Zhang and M. Yang, "Real-time compressive tracking", In European Conference on Computer Vision, Florence, Italy, **(2012)**, pp. 864-877.
[15] Q. Wang, F. Chen, W. Xu, and M.-H. Yang., "Object tracking via partial least squares analysis", IEEE Transactions on Image Processing, vol. 21, no. 10, **(2012)**, pp. 4454–4465.
[16] W. Zhong, H. Lu and M. –H. Yang, "Robust object tracking via sparsity-based collaborative model", In Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, RI, USA, **(2012)**, pp. 1838-1845.
[17] Q. Wang, F. Chen, J. Yang, W. Xu and M.-H. Yang, "Transferring visual prior for online object tracking", IEEE Transactions on Image Processing, vol. 21, no. 7, **(2012)**, pp. 3296–3305.
[18] X. Li, C. Shen, Q. Shi, A. R. Dick and A. van den Hengel, "Non-sparse linear representations for visual tracking with online reservoir metric learning", In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, **(2012)**, pp. 1760–1767.
[19] C. Bao, Y. Wu, H. Ling and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach", In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, **(2012)**, pp. 1830–1837.
[20] T. Zhang, B. Ghanem, S. Liu and N. Ahuja, "Low-rank sparse learning for robust visual tracking", In European Conference on Computer Vision, **(2012)**, pp. 470–484.
[21] X. Jia, H. Lu and M.-H. Yang, "Visual Tracking via Adaptive Structural Local Sparse Appearance Model", In Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, RI, USA, **(2012)**, pp. 1822-1829.
[22] T. Zhang, B. Ghanem, S. Liu and N. Ahuja, "Robust Visual Tracking via Multi-Task Sparse Learning", In Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, RI, USA, **(2012)**, pp.2042-2049.
[23] Y. Peng, A. Ganesh, J. Wright, W. Xu and Y. Ma, "RASL: Robust Alignment by Sparse and Low-rank Decomposition for Linearly Correlated Images", In Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, San Francisco, CA, **(2010)**, pp. 763-770.
[24] A. Yilmaz, O. Javed and M. Shah, "Object tracking: A survey", ACM Computing Surveys, vol. 38, no. 4, **(2006)**, pp. 1-45.
[25] Q. Zhou, H. Lu and M. Yang, "Online multiple support instance tracking", In Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition and Workshop , IEEE, Santa Barbara, CA, **(2001)**, pp. 545-552.

[26] N. Wang and D.-Y. Yeung, "Learning a Deep Compact Image Representation for Visual Tracking", In Proceedings of Twenty-Seventh Annual Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, USA, **(2013)**, pp. 5-10.
[27] J. Mairal, F. Bach, J. Ponce and G. Sapiro, "Online learning for matrix factorization and sparse coding", Journal of Machine Learning Research, no. 11, **(2010)**, pp. 19-60.
[28] J. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The pascal visual object classes(voc) challenge", International Journal of Computer Vision, vol. 88, no. 2, **(2010)**, pp. 303-338.

Authors

Yang Liu, she received her M.S. degree from the Northeastern University in China in 2004. She holds the position of Associate Professor at Shenyang Aerospace University. She is a doctoral student at the Nanjing University of Aeronautics and Astronautics. Her research interests include vision analysis and pattern recognition.

Yibo Li, he received his M.S. and Ph.D. degrees from the Nanjing University of Aeronautic and Astronautics and Northeastern University, in 1986 and 2003, respectively. Since 1999, he holds the position of Full Professor at Shenyang Aerospace University. He has published over 100 technical research papers and books. More than 30 research papers have been indexed by SCI/EI. His research interests include vision analysis and pattern recognition.

Xiaofei Ji, she received her M.S. and Ph.D. degrees from the Liaoning Shihua University and University of Portsmouth, in 2003 and 2010, respectively. From 2013, she holds the position of Associate Professor at Shenyang Aerospace University. She is the IEEE member, has published over 20(indexed by SCI/EI) technical research papers. Her research interests include vision analysis and pattern recognition. She is the leader of National Natural Science Fund Project (Number: 61103123).

Yangyang Wang, she received her M.S. degree from the Shenyang Institute of Aeronautical Engineering, in 2006. She is currently a graduate student studying for Ph.D. degree in the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics. She has published over 10 technical research papers. Her research interests include vision analysis and pattern recognition.

