International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.3 (2014), pp.407-420
http://dx.doi.org/10.14257/ijsip.2014.7.3.33

Robust Object Tracking with Occlusion Handling based on Local Sparse Representation

Hainan Zhao, Xuan Wang and Meng Liu

Computer Application Research Center, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China
hainan.hh@gmail.com, xuanwang@insun.hit.edu.cn, liumeng@sdu.edu.cn

Abstract

Sparse representation has been successfully applied to visual tracking to find the target with the minimum reconstruction error from the target templates subspace. Traditional sparsity-based trackers handle corruptions and occlusions of the observation by introducing a set of trivial templates. However, the performance is not so satisfactory in practice. It is because the trivial templates unable to model heavy occlusions effectively, and the likelihood computation and the template update processes do not take full advantage of the occlusion information. In this paper, we propose a novel tracking method taking advantage of local sparse representation to detect occlusions during the tracking sequence. In our method, the target is divided into local patches. We analyze the spatial distribution of the samples employed by the local sparse representation, and determine the occlusion state for each patch respectively. The occluded patches are disregard, only the unoccluded ones are considered for reconstruction and likelihood computation. In addition, a dynamic template update strategy with occlusion handling is introduced to alleviate the drift problem. Experiments on challenging video sequences demonstrate that the proposed tracking algorithm performs favorably against several state-of-the-art methods.

Keywords: *visual tracking, occlusion handling, sparse representation, local patch, $\ell 1$ minimization, template update*

1. Introduction

Visual tracking has been an important topic in the field of computer vision for decades. It plays a key role in numerous applications such as intelligent surveillance, activity analysis, aided navigation, *etc*. The main challenge in designing a robust visual tracking algorithm is the inevitable appearance variations of the target observation which are mainly caused by partial occlusion, illumination change, background clutter, viewpoint variation and so on. Among many factors, occlusion is one of the most critical issues since it is difficult to be modeled but can greatly influence the tracking result. To deal with occlusions, a large number of effective tracking methods have been developed including statistical analysis method [1], fragments-based method [2] and spatiotemporal context based method [4].

Sparse representation offers a new insight for solving object occlusion problem. Mei *et al.*, [11] formulate tracking as a sparse approximation problem in a particle filter framework, and deal with corruptions and occlusions by introducing a set of trivial templates. However, the trivial templates are unable to model large occlusion effectively. The images from the occluded regions may also get small reconstruction errors, so that might be mistaken as the tracking result. Moreover, the likelihood computation and template update processes do not take full use of the occlusion information captured in the sparse coefficients, the tracking

ISSN: 2005-4254 IJSIP

result with large occlusion might be added to the template set. Ambiguities accumulate and finally cause tracking failure. A modified method is proposed in [15], which detects occlusion through learning with observation likelihoods. However, the occlusion classifier needs to be trained in advance, and the training and testing should be performed in the similar environment to keep tracking reliability. Those restrictions limit its applicability.

In this paper, we propose a novel tracking algorithm which takes advantage of local sparse representation to deal with occlusions. In our approach, a sample set which contains both target and background templates extracted from a sub-image centered around the target is constructed. We use this sample set to capture the local visual information of both the target and the possible occluders. In each tracking frame, the target region is divided into multiple local patches. We reconstruct each patch using the sample set and explore the spatial information captured by its sparse coefficient vector for occlusion detection. An occlusion mask is then constructed according to the occlusion status of all these patches, which is used to ensure that only the unoccluded patches are involved in the likelihood computation procedure. In additional, we introduce a dynamic template update strategy with the consideration of occlusion to adapt to the variations of the target appearance. The rarely used templates are replaced with the tracking result templates or their recovered templates to alleviate the drift problem.

The rest of the paper is organized as follows. In the next section related work are summarized. Our occlusion detection method based on local sparse representation is described in Section 3. The proposed tracking algorithm and the dynamic template update strategy are presented in Section 4. We illustrate experimental results with both qualitative and quantitative evaluations in Section 5. Finally, concluding remarks are given in Section 6.

2. Related Work

Appearance model is usually the essential component in all tracking methods. To deal with the appearance variations of the target object, many sophisticated object representation methods have been proposed, which can be generally categorized into either discriminative [6, 7, 8] or generative [2, 10, 11] methods.

Discriminative methods formulate tracking as a binary classification problem. The candidate which can be best separated from the background is taken as the tracking result. Avidan, *et al.*, [6] propose an ensemble tracking framework, in which a confidence map is constructed using an ensemble of weak classifiers to separate pixels that belong to the object from ones that belong to the background. The peak of the map is considered as the new position of the object. Babenko, *et al.*, [7] use multiple instance learning (MIL) instead of traditional supervised learning to learn a discriminative model for tracking. A discriminative appearance model based on super pixels is introduced in [8]. It facilitates the tracker to distinguish between target and background. These discriminative tracking methods aim to construct a good appearance model for effectively separating the object from background. How to correctly label the samples for training and updating the classifier is still a challenge. Generative methods formulate the tracking problem as searching for the region most similar to the target appearance. One of the critical issues for generative methods is how to make the tracker adapt to the inevitable appearance variation of the target. An online algorithm in [9] incrementally learns a low dimensional eigenspace representation to reflect appearance changes of the target [10]. decomposes the observation model into multiple basic observation models. Each basic observation model covers a specific appearance variation of the target, so that the compound observation model can be robust to combinatorial appearance variation. The target template is represented by multiple image patches to handle partial occlusion and

pose change in [2]. Moreover, algorithms take advantage of both generative and discriminative models are proposed in [16, 23].

Recently sparse representation has been extensively studied and successfully applied to computer vision and pattern recognition, *e.g.*, face recognition [20], super-resolution [21] and image in painting [13]. Motivated by the work in [20], Mei *et al.*, apply sparse representation to visual tracking [11], and further extend the tracker by combining the tracking with recognition simultaneously in [12]. In [15-17], local sparse representation is used to effectively manage the appearance changes of the object over time. Han, *et al.*, [22] propose a sample-based adaptive sparse representation (AdaSR) method. Tracking is implemented by searching for the regions holding the most similar AdaSR to that of the target. Wang, *et al.*, [19] propose an online object tracking algorithm, which takes advantage of both principal component analysis (PCA) algorithm and sparse representation scheme to learn an adaptive appearance model.

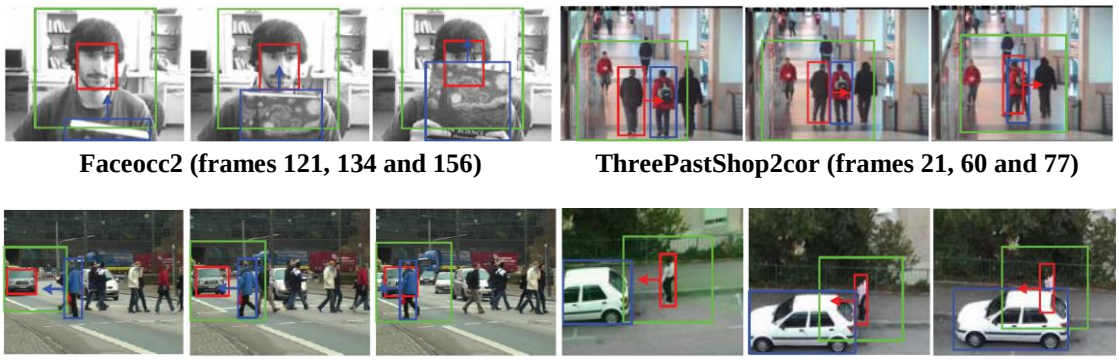
Sharing philosophies with works above, we develop an effective object representation method which takes advantage of both generative mode and discriminative model of the object. The generative model is constructed from the holistic template of the object to adapt to the appearance variations during tracking process, while the discriminative model takes use of the local sparse representation of the object to detect occlusions. This method ensures that the occluded patches are excluded from reconstruction and likelihood computation. Compared with sparsity-based models [11, 12, 14], our method maintains local appearance information and therefore can effectively manage occlusion problem. Our work bears some similarity to [15] in the use of local sparse representations. Yet we construct the occlusion classifier online rather than train the classifier in advance, which improves the robustness and adaptability of the tracker. Furthermore, we improve the template updating operation according to the occlusion status of the tracking result to prevent pollution of the template set and get satisfactory results.

3. Occlusion Detection based on Local Sparse Representation

We now describe how to design the detector for occlusion and discuss the important characteristics related. In our method, the information hides in the spatiotemporal context of the tracking scene is explored. We use this information to design a general and robust algorithm for detecting occlusions adaptively. We first detail the local sparse representation as the basic of the detector, and then explain the classification rule of occluded and unoccluded local patches.

3.1. Local Sparse Representation

Local templates (fragments) based methods have been verified as a king of effective techniques for handling occlusions [2, 16]. The key issue of these methods is how to properly label the positive and negative samples to discriminate the target from the occluder. Missing updated with the occluder patches as positive samples, always leads to the template drift and tracking failure. By analyzing the causes of occlusions on several video sequences shown in Figure 1, it is not difficult to discover that the possible occluders always appeared in the former several frames before occlusion occurs. We combine spatiotemporal context of the tracking scene and sparse representation to model the occlusion region, and present a kernel based method to adaptively label the positive and negative samples. This method enhances the robustness and reliability of the occlusion classifier. Details will follow.



Faceocc2 (frames 121, 134 and 156) **ThreePastShop2cor (frames 21, 60 and 77)**

Tud-crossing-sequence (frames 10, 12 and 31) **Woman (frames 81, 104 and 120)**

Figure 1. Illustration of Occlusion Situations: Target, Possible Occluder and Sample Window are represented by Red, Blue and Green Bounding Boxes, Respectively; Arrow Denotes Moving Direction

When the state of the target is estimated in each frame, the target image region is divided into 4×4 local image patches, as shown in Figure 2. Then each local patch is normalized to the same size and stacked to a vector $\chi_i \in \square^{H\times 1}$, where H is the size of the normalized patch. To estimate if a patch is occluded, a sample set is constructed based on a sample window, which is a rectangle region centered around the target. Each template in the set is defined as a sub rectangle region of the sample window specified by $d_j = (c_j, \omega_j, s_j)$, where c_j is the center coordinates which is generated according to uniformly distribution inside the sample window, ω_j is the rotation angle, it is set to be equal with that of the target. s_j is the size which is variably set to 0:9, 1:0 and 1:1 times of H to manage the scale variation of the occluder. All the templates are normalized and turned into vectors in the same way as the local patch of the target. With the sparsity assumption, each local patch of the target can be sparsely represented as the linear combination of the templates of the sample set by solving the regularized ℓ1 minimization as

$$b_i^* = \arg\min_{b_i} \| \chi_i - Db_i \|_2^2 + \lambda \| b_i \|_1, \quad s.t. b_i \mu\ 0 \qquad (1)$$

Where $D = [d_1, d_2, ..., d_m] \in \square^{H\times m}$ denotes the sample set, m is the number of templates in the set and is set to 150 in our experiment to balance between effectiveness of modeling occlusion and computational efficiency. $b_i \in \square^{m\times 1}$ is the sparse coefficient vector of the *i*-th local patch, and λ is a regularization constant to balance reconstruction error and sparsity. $b_i \mu\ 0$ Indicates all the elements of b_i are nonnegative. By this way, each local patch of the target region is sparsely represented by the most representative templates of the sample set. The larger the coefficient value, the more relevance exists between the corresponding template and the local patch.

3.2. Kernel based Occlusion Detector

It is reasonable to assume that the unoccluded patches of the target can be better represented by the linear combination of templates from target region, while the occluded patches can be better represented by the span of templates from the region belongs to possible

occluders. What should be pointed out in advance is that the sample set is sampled at a proper former frame before the occlusion occurs, which is to ensure the templates belong to the occluder region can keep a spatial distance with the templates belong to the target region. The leads depends on the frame frequency and the relative approaching velocity of the target and the occluder, and 3 to 9 frames ahead is a empirical choice in most of the real world video scenes. Taking the sampling strategy above, the templates from the possible occluder will appear in the outer region of the sample window, while the templates from the target will appear in the central region. Thus, a weight value of occlusion status for each local patch of the target is generated by

$$\sigma_i = \frac{1}{m}\sum_{j=1}^{m} k(\| \frac{c_j - c_{tar}}{h} \|^2) b_i^{*j} \quad (2)$$

where b_i^{*j} is the j-th element of sparse coefficient vector b_i^*, c_{tar} is the estimated center coordinates of the target, c_j corresponds to the location of the j-th template in the sample set, h is bandwidth used to adapt to the size of the target region, and $k(x)$ is an isotropic kernel profile. In order to make $\sigma_i = 0$ as the cut-off point of occluded patches and unoccluded patches, we extend the Epanechnikov kernel symmetrically by negative weights

$$\mathrm{K}(u) = \frac{3}{4}(1-u^2)1_{\{|u|\le d\}} \quad (3)$$

Where d is a constant accounting of width proportion of sample window and target region. Figure 2 illustrates the spatial distribution of the templates employed by the sparse representation of the occluded and unoccluded target local patches. Examples of occluded and unoccluded local patches are represented by the blue and red bounding boxes respectively in Figure 2(a). Figure 2(b) and (c) show their sparse coefficient vectors. We can see that both of the local patches can be sparsely represented by templates from the sample set, but the spatial distributions of the templates employed are different. By converting the 1D coefficient vector to a 2D image according to the location of the templates, we get an intuitive comparison of spatial distribution of these templates employed in two representations. As shown in Figure 2(d) and (e), the unoccluded patch is represented by templates from the target region with large coefficients (indicated by the red bars). To the contrary, the occluded patch is sparsely represented by templates from the occluder region (indicated by the blue bars). We get a weight image by weighting each coefficient value in the 2D image using kernel function $\mathrm{K}(x)$. In this weight image, the templates well represent the unoccluded patch get positive weight values, and negative for templates associate with the occluded patch. The kernel function increases the absolute weight values of templates locating at the central and boundary areas of the sample window, and decreases the absolute weight values of the ones locating at the border of target region. Through this way, we improve the robustness of the occlusion classifier.

After computing *Eq.* 2, the unoccluded local patches get positive values, and the occluded ones get negative values. We assign the pixels belong to the unoccluded patches a value of 1, and the pixels belong to the occluded patches a value of 0. Thus we get an occlusion map denoted by $M_o \in \square^{w \times h}$ which indicates the occlusion status of the integral target region, where w is the width of the target region and h is the height.

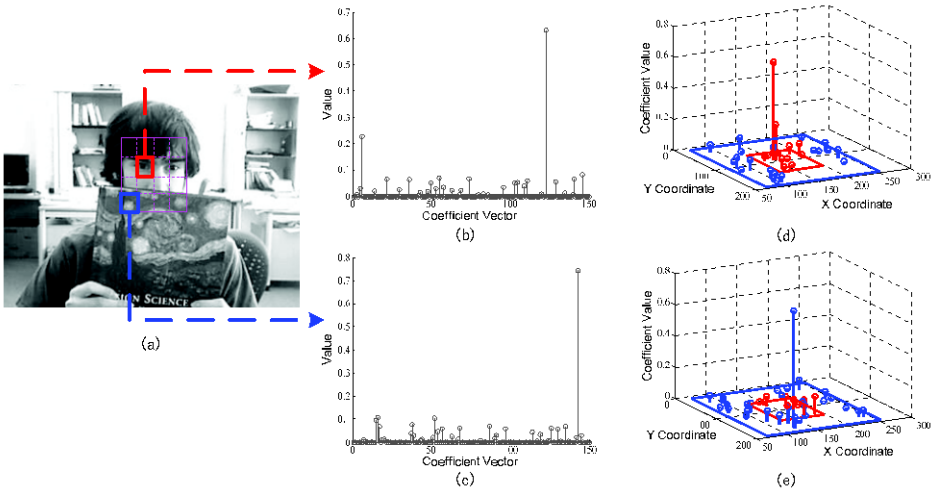


Figure 2. Illustration of Spatial Distribution of the Templates Employed by the Sparse Representations of the Occluded and Unoccluded Target Local Patches

4. Proposed Tracking Algorithm

Our object tracking algorithm is carried out within the Bayesian inference framework. Given the observation set of target $Z_t = \{z_1, z_2, ..., z_t\}$ up to the *t*-th frame, we estimate the target state variable x_t by maximizing the posteriori probability over N samples at frame *t* by

$$x_i^* = \arg\max_{x_t^i} p(x_t^i \mid Z_t),\ i = 1, 2, ..., np \qquad (4)$$

Where x_t^i indicates the state of the *i*-th sample at the *t*-th frame. The posteriori probability $p(x_t \mid Z_t)$ (we drop the sample index *i* for generality) can be estimated recursively by

$$p(x_t \mid Z_t) \propto p(z_t \mid x_t) \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid Z_{t-1}) dx_{t-1} \qquad (5)$$

Where $p(x_t \mid x_{t-1})$ is the dynamic (motion) model between two consecutive states? Let x_t be the six-dimensional parameter vector for affine transformation. We model the transformation of each parameter independently by a scalar Gaussian distribution between two consecutive frames. Then the dynamic model can be represented by a Gaussian distribution $p(x_t \mid x_{t-1}) = N(x_t; x_{t-1}, \psi)$, where ψ a diagonal covariance matrix is whose elements are the variances of the affine parameters. $p(z_t \mid x_t)$ Is the observation model which denotes the likelihood of the observation z_t at candidate state x_t ? We formulate $p(z_t \mid x_t)$ using the reconstruction error in the sparse representation elaborated later. One of the contributions of this work is that we handle partial occlusion in an explicit and effective way in the construction of the observation model $p(z_t \mid x_t)$. Figure 3 depicts the main components of our tracking algorithm, which consists of three main parts: occlusion detection, tracking with occlusion handling and the adaptive updating strategy. The detail will be discussed later in this section.

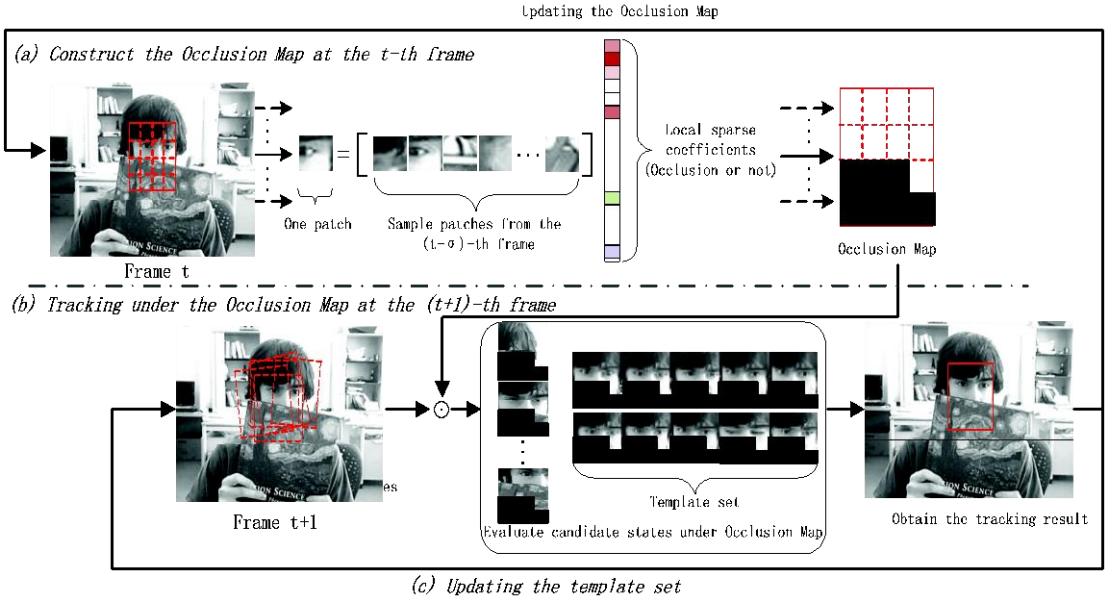


Figure 3. Components of the Proposed Tracking Algorithm

4.1. Tracking with Occlusion Handling

Be different from the tracker in [11], only the unoccluded patches are used for reconstruction and likelihood computation in our method. Let $y_i \in \mathbb{R}^d$ be a target candidate, and $T = [t_1 ... t_n] \in \mathbb{R}^{d \times n}$ be the set of target templates constructed in the same way as the one in the original $\ell 1$ minimization tracking [11]. We apply the occlusion map to the target candidate y_i and all the target templates t_i in T simultaneously in order to exclude the occluded patches. Note the occlusion map M_o constructed in frame t is utilized to detect the occlusion status of target at frame t+1. Let $y_i \in \mathbb{R}^k$ ($k \le d$) denotes the target candidate with only unoccluded patch, and $T \in \mathbb{R}^{k \times n}$ denote the corresponding target template set, the sparse coefficient vector of y_i is given by

$$a_i^* = \arg\min_{a_i} \| Ta_i - y_i \|_2^2 + \lambda \| a_i \|_1 , \ s.t. \ a_i \mu 0 \qquad (6)$$

Where $a_i \in \mathbb{R}^n$. If there is no occlusion, $k = d$. Given a set of target candidates $Y = \{y_1, ..., y_{np}\}$ generated by the particle filter framework in each frame, the observation likelihood of candidate y_i is derived from the reconstruction error of y_i as

$$\varphi_i^* = \frac{1}{\kappa} \exp\{-\eta \| Ta_i^* - y_i \|^2\} \qquad (7)$$

where η is a constant that controls the shape of the Gaussian kernel, and κ is a normalization factor. For tracking at time t, the candidate with the maximum observation likelihood is chosen as the tracking result. By solving *Eqs.* 6 and 7, our tracker removes the influence of the occluded patches in likelihood computation, which improves the reliability and accuracy of the tracking result.

4.2. Template Update

Since the appearances of the target may change significantly due to factors such as illumination variation, pose change, viewpoint variation, *etc.*, it is necessary to update the template for adapting to these appearance changes during the tracking process. Many approaches have been proposed for template update. Mei and Ling [11, 12] replace the rarely used template in the template set when none of the template is similar with the tracking result. This update approach does not take any steps to prevent tracking result with large occlusion from being added to the template set. Jia *et al.*, [17] use the reconstructed image based on only PCA basis vectors to replace the template which is later added to the template set. This method is likely to fail when the tracking result is severely occluded or the target appearance changes significantly. Using defective samples to update template set or undeservedly replacing representative template is the leading cause of drift problem. A good template update strategy can not only capture the appearance change of target, but also preserve the common information of the target in each frame. Our template update approach takes into consideration the occlusion status of the tracking result to prevent template with large occlusion from being added to the template set.

Table 1. Algorithm for Template Update

Input: Newly tracking result y^*, the corresponding coefficient vector α^* and observation likelihood φ^*, occlusion weight values of all local patches $\sigma=[\sigma_1,\sigma_2,...,\sigma_{16}]$, template set T, weight vector at the previous Frame w^{t-1} and predefined thresholds γ, τ_1, τ_2.

Output: New template set T, new weight vector w^t

1: Update the weights vector by Eq. 8
2: If $\varphi^* < \gamma$ then
3: Compute the occlusion ratio $\delta \leftarrow \sum_{i=1}^{16} \mathrm{f}(\sigma_i)/16$ where $\mathrm{f}(\sigma_i) = \begin{cases} 0 & \sigma_i \geq 0 \\ 1 & otherwise \end{cases}$
4: $j_0 \leftarrow \arg\min_{1\leq j\leq n} \omega_j^t$
3: If $\delta < \tau_1$ then
4: $t_{j_o} \leftarrow y^*$ /*replace an old template directly*/
5: else if $\tau_1 \leq \delta \leq \tau_2$ then
6: $y^* \leftarrow \mathrm{avg}(y^*, T, \sigma)$ /* replace the occluded patches */
7: $t_{j_o} \leftarrow y^*$ /*replace an old template using the recovered sample */
8: end if
9: end if
10: end if

The template update algorithm is summarized in Table 1. We update the template only if the observation likelihood φ^* (calculated by *Eq.* 7) of the tracking result y^* is smaller than a certain threshold γ, which indicates y^* can't be well represented by current template set T. We use the ratio δ of the number of occluded patches to the number of all patches inside the target region to measure the degree of occlusion. Two thresholds τ_1 and τ_2 are predefined to describe the degree of occlusion ($\tau_1 = 0.1$ and $\tau_2 = 0.6$, in our experiments). Three kinds of updating operations [19] are adopted according to the occlusion ratio δ. If $\delta < \tau_1$, we directly update the template with the tracking result

sample. If $\tau_1 \le \delta \le \tau_2$, it indicates that the target is partially occluded. We replace the occluded patches by the average of the corresponding parts of the current templates, and use this recovered sample for updating. If $\delta > \tau_2$, it indicates that the tracking result target is severely occluded, in this case we do not trust the target sample and discard it without updating. In order to keep the stable templates, and exclude the rarely used ones, we introduce an importance weight for each template as [11] does, which is given by

$$w^t = cw^{t-1} * \exp(\alpha^*) \quad (8)$$

Where $w^t = [\omega_1^t, \omega_2^t, ..., \omega_n^t] \in \mathbb{R}^n$, n is the number of templates, ω_j^t denotes the importance weight of the *j*-th template at frame t. *c* is a normalization constant which makes $\sum_{j=1}^{n} \omega_j^t = 1$ .The weight of each template increases when the template gets a large coefficient in the target reconstruction and decreases otherwise.

5. Experiments

We evaluate the performance of the proposed algorithm on eight challenging image sequences. These sequences cover different kinds of tracking objects with various occlusion situations. The proposed approach is compared with six state-of-the-art tracking methods including incremental visual tracking (IVT) method [9], ℓ1 tracker (ℓ1) [11], multiple instance learning (MIL) tracker [7], the online AdaBoost method (OAB) [24], P-N learning (PN) [5] and tracking method with sparse prototypes (SRPCA) [19]. For fair comparison, we use the source or binary codes provided by the authors with tuned parameters for best performance.

Our tracker is implemented in MATLAB, which runs at 2.8 frames per second (fps) on a PC with Intel Core i7-3770 CPU (3.4 GHz) with 16 GB memory. The target image observation is normalized to 32×32 pixels, and the size of the normalized local patch H is set to 64. The regularization constant λ in *Eqs*. 3 and 6 is set to 0.01 in all experiments. As a trade-off between computational efficiency and effectiveness, 600 particles are used and the template set is updated every 5 frames. Only gray scale information is used in our experiments. For each sequence, the location of the target object is manually labeled in the first frame. Both qualitative and quantitative evaluations are presented in this section.

5.1. Qualitative Evaluation

Results from two face tracking sequences with partial occlusion, as well as large pose variation are shown in Figure 4(I) and (II). For the Faceocc2 sequence, many trackers drift apart from the target or do not scale well when the face is heavily occluded. Although the MIL tracker is able to track the target object, it is not able to estimate the in-plane rotation due to its design. Our and the SRPCA methods are able to track the target accurately throughout the video sequence. Our tracker performs well especially when partial occlusion and in-plane rotation occur simultaneously (frame 491). This can be attributed to our occlusion detection method that alleviates the influence of occlusions. For the Girl sequence, the target girl's face undergoes in-plane rotation and occlusion from a man's face passing in front of it. The IVT method drifts away quickly around frame 119. This result demonstrates that the IVT method based on PCA subspace representation is sensitive to in-plane rotation. The ℓ1, SRPCA and our methods successfully track the target, while other trackers drifts

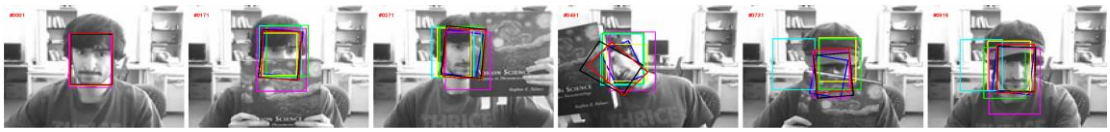
apart when partial occlusion occurs. Our tracker obtains better tracking accuracy in some of the frames (428 and 474).

Results of different algorithms in surveillance videos are shown in Figure 4(III) and (IV). These videos are challenging as they contain partial occlusion, as well as scale change and similar objects. In the PETS01_Human sequence, our target object is a man with white shirt walking through the square. It can be observed that all trackers except ours fail to track the target around frame 96 due to heavy occlusion. The PN algorithm is equipped with a re-initialization mechanism based on object detection with global search, therefore it may be able to track the object again by chance when the target object reappears after occlusion (frame 136), and however it loses the target finally due to small size. It should be noted that our tracker shows some target drifting when the target man is blocked by the lamp (frame 264) or the van (frame 292). This result is due to the fact that the target becomes smaller and smaller, hence the local patch images may not capture sufficient visual information to represent objects for occlusion detection. Even so, our tracker successfully tracks the target throughout the video sequence. For ThreePastShop2cor sequence, the target is occluded by two people successionally and one of them is similar in color and shape to the target. Most trackers fail during the first occlusion occurs (frame 62 to 92) due to the heavy occlusion. The sparsity-based trackers perform well before frame 92; however the $\ell 1$ tracker loses the target after frame 124, and locks on the people with similar appearance after occlusion. This can be explained by that the simple trivial template set employed by the $\ell 1$ tracker models occlusion less effectively, moreover, the update method of it takes new image observations for update without factoring out occlusion. In contrast, the SRPCA and our trackers achieve stable performance in the entire sequence, since the update schemes of the both trackers do not introduce heavy occlusion which is the main reason for drift problem.

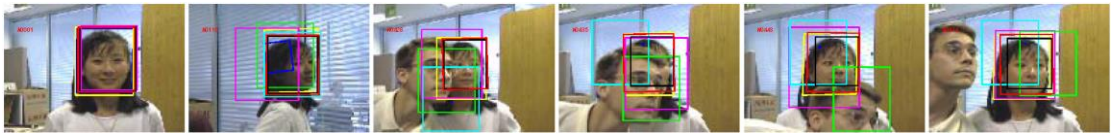
Figure 4(V) and (VI) illustrate the tracking results using the Woman and Bolt sequences. As the objects undergo long-time partial occlusion accompanied with non-rigid pose variation, it is difficult to predict their locations. For the Woman sequence, based on the local patches occlusion detection method and adaptive template update strategy, our tracker focuses more on the upper body which remains almost the same though the lower body changes a lot or is heavily occluded. It can successfully track the target throughout the entire sequence. For the Bolt sequences, PN tracker drifts away quickly as the target pose changes, since it relies heavily on the visual information in the first frame to re-detect the object. Other trackers lost the target successively after frame 71. Note that the SRPCA method keeps track of the target object at the beginning of occlusion (frame 92), but losts the target finally due to complex background. Our tracker drifts apart somewhat around frame 154 due to heavy occlusion as well as large pose variation. However, it locks on the target again when the target reappears after occlusion. This can be attributed to that our method updates appearance change correctly especially when heavy occlusion occurs.

Results from the football sequence are shown in Figure 4(VII), in which the object undergoes partial occlusion in severe cluttered background. The trajectory of the target is hijacked by another football player wearing a similar helmet to the target as the two players collided with each other at frame 288. Our method overcomes this problem and successfully tracks the target.

We last test our algorithm in the Tud-crossing sequence, and the sample results are illustrated in Figure 4(VIII). The challenge in this sequence is multiple occlusions come from various directions. The PN tracker is able to track the target, but with higher tracking errors and lower success rate. The SRPCA and our methods track the target successfully and maintain target locations stably even with multiple occlusions by pedestrians. Other methods are distracted by the occlusions significantly.



(I) Faceocc2 (frames 1, 171, 271, 491, 731 and 816)



(II) Girl (frames 1, 119, 428, 435, 443 and 474)

(III) PETS01_Human (frames 1, 96, 136, 264, 292 and 490)

(IV) ThreePastShop2cor (frames 1, 62, 92, 124, 143 and 350)

(V) Woman (frames 1, 42, 120, 214, 384 and 552)

(VI) Bolt (frames 1, 71, 92, 154, 164 and 208)

(VII) Football (frames 1, 60, 123, 288, 337 and 362)

(VIII) Tud-crossing-sequence (frames 1, 33, 51, 75, 95 and 127)

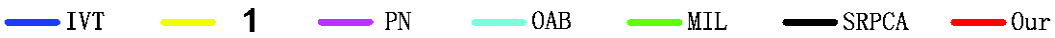


Figure 4. Samples of Tracking Results

5.2. Quantitative Evaluation

Performance evaluation is an important issue that requires sound criteria in order to fairly assess the strength of tracking algorithms. We employ two typical evaluation criteria to quantitatively assess the performance of these trackers. The first one is center location error which is approximated by the distance between the central position of the tracking result and that of the manually labeled ground truth. Table 2 summarizes the results in terms of average center location error. We note that although PN tracker is able to relocate on the target during tracking, it is easy to lose the target completely for some frames in most of the test sequences. Thus, we only show the center location errors for the sequences that PN tracker can keep track all the time. The second criterion is the tracking overlap rate which indicates stability of each algorithm as taking size and pose of the target object into account. Given the tracked bounding box ROI_{TK} and the ground truth bounding box ROI_{GT}, the overlap rate is defined by the PASCAL VOC [18] criterion, $score = \frac{area(ROI_{TK} \cap ROI_{GT})}{area(ROI_{TK} \cup ROI_{GT})}$. Figure 5 shows the overlap rates of each tracking algorithm for all the sequences. Overall, the proposed tracker performs favorably against state-of-the-art methods.

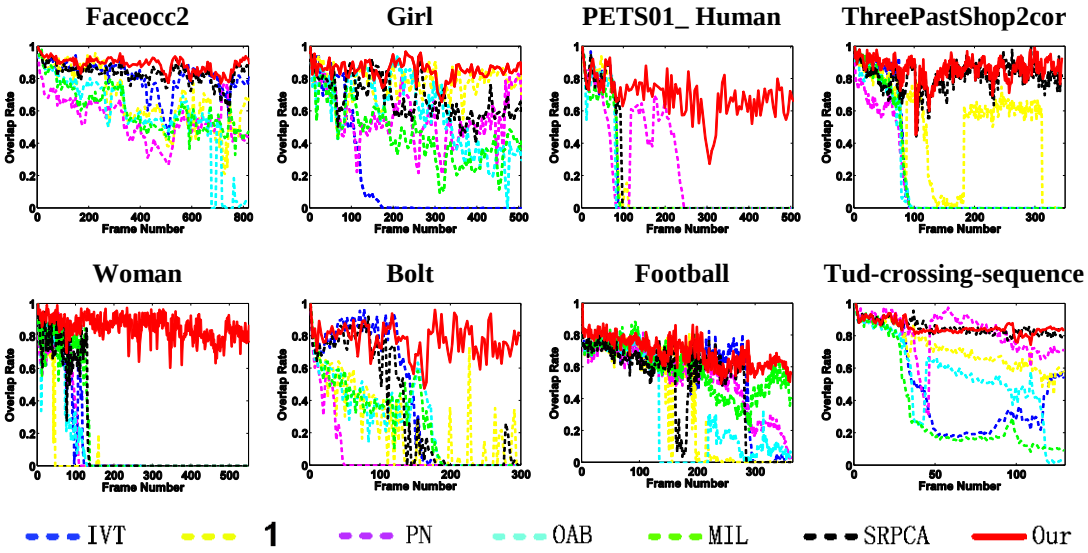


Figure 5. Overlap Rate Evaluation for Eight Video Sequences

6. Conclusion

This paper presents a simple but effective method to detect occlusions for visual tracking. We provide an effective way to model occlusions via local sparse representation. We explicitly take occlusion into account for likelihood computation, this helps the tracker locate the target more accurately and be less insensitive to occlusion. Our occlusion detection method integrated with a adaptive template update scheme prevents the tracking result with heavy occlusions from being added to the target template set. Preventing an incorrect update reduces tracking failure. Experimental results compared with several state-of-the-art methods on challenging sequences demonstrate the effectiveness and robustness of the proposed algorithm. However, the proposed method requires high computational cost due to calculations for

ℓ1 minimization. Furthermore, the computational cost grows proportionally as the number of local patches increases. The large computational cost prevents the tracker from being used in a real time system. We expect further future study along this direction.

Table 2. Average Center Location Error (Pixels). The Best Two Results are Shown in Bold and Italic Fonts

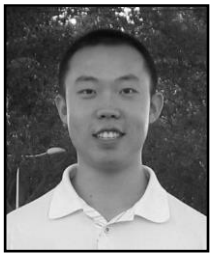
Image sequence	IVT	ℓ1	MIL	OAB	PN	SRPCA	Ours
Faceocc2	4.9	11.1	14.3	19.2	14.3	**4.0**	*4.6*
Girl	13.4	3.3	13.9	9.6	9.3	*3.3*	**2.7**
PETS01_ Human	256.4	*237.2*	254.5	259.2	--	247.6	**4.7**
ThreePastShop2cor	65.4	12.4	71.9	92.2	--	*5.5*	**3.3**
Woman	190.7	198.5	*126.7*	189.7	--	131.4	**4.9**
Bolt	57.7	41.1	38.6	*36.8*	--	37.86	**9.1**
Football	18.5	33.6	*10.1*	80.9	12.3	37.3	**6.4**
Tud-crossing-sequence	42.0	10.6	56.9	22.9	8.4	**4.1**	*8.1*
Average	81.1	68.5	73.4	88.8	--	*58.9*	**5.5**

References

[1] B. Han and L. Davis, "On-line density-based appearance modeling for object tracking", Proceedings of the IEEE 10th International Conference on Computer Vision, China, **(2005)**, pp. 1492–1499.
[2] A. Adam, E. Rivlin and I. Shimshoni, "Robust fragments-based tracking using the integral histogram", Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, USA, **(2006)**, pp. 798–805
[3] K. Cannons, "A review of visual tracking", Tech. Rep. CSE-2008-07, Dept. Comput. Sci. Eng.,York Univ., Canada, **(2008)**.
[4] M. Yang, Y. Wu and G. Hua, "Context-aware visual tracking", IEEE Trans. PAMI, vol. 31, no. 7, **(2009)**, pp. 1195–1209.
[5] Z. Kalal, J. Matas and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints", Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition, USA, **(2010)**, pp. 49–56.
[6] S. Avidan, "Ensemble tracking", IEEE Trans. PAMI, vol. 29, no. 2, **(2007)**, pp. 261–271.
[7] B. Babenko, M. H. Yang and S. Belongie, "Visual tracking with online multiple instance learning", Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, USA, **(2009)**, pp. 983–990.
[8] S. Wang, H. Lu, F. Yang and M. H. Yang, "Superpixel tracking", Proceedings of the IEEE 13th International Conference on Computer Vision, Spain, **(2011)**, pp. 1323–1330.
[9] D. A. 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.
[10] J. Kwon and K. M. Lee, "Visual tracking decomposition", Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition, USA, **(2010)**, pp. 1269–1276.
[11] X. Mei and H. Ling, "Robust Visual Tracking using ℓ1 Minimization", Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, USA, **(2009)**, pp. 1436–1443.
[12] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation", IEEE Trans. PAMI, vol. 33, no. 11, **(2011)**, pp. 2259–2272.
[13] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang and S. Yan, "Sparse Representation for Computer Vision and Pattern Recognition", Proceedings of the IEEE, vol. 98, no. 6, **(2010)**, pp. 1031–1044.
[14] X. Mei, H. Ling, Y. Wu, E. Blasch and L. Bai, "Efficient Minimum Error Bounded Particle Resampling ℓ1 Tracker With Occlusion Detection", Image Processing, IEEE Transactions. on, vol. 22, no. 1, **(2013)**, pp. 2661–2675.
[15] S. Kwak, W. Nam, B. Han and J. H. Han, "Learning Occlusion with Likelihoods for Visual Tracking", Proceedings of the IEEE 13th International Conference on Computer Vision, Spain, **(2011)**, pp. 1551–1558.
[16] W. Zhong, H. Lu and M. H. Yang, "Robust Object Tracking via Sparsity-based Collaborative Model", Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, USA, **(2012)**, pp. 1838– 1845.

[17] X. Jia, H. Lu, and M. H. Yang, "Visual Tracking via Adaptive Structural Local Sparse Appearance Model", Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, USA, **(2012)**, pp. 1822–1829.
[18] M. Everingham, L. Van Gool, C. 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.
[19] D. Wang, H. C. Lu and M. H. Yang, "Online Object Tracking With Sparse Prototypes", Image Processing, IEEE Transactions on, vol. 22, no. 1, **(2013)**, pp. 314–325.
[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, "Robust face recognition via sparse representation", IEEE Trans. PAMI, vol. 31, no. 2, **(2009)**, pp. 210–227.
[21] J. Yang, J. Wright, T. S. Huang and Y. Ma, "Image super-resolution via sparse representation", Image Processing, IEEE Transactions on, vol. 19, no. 11, **(2010)**, pp. 2861–2873.
[22] Z. J. Han, J. B. Jiao, B. C. Zhang, Q. X. Ye and J. Z. Liu, "Visual object tracking via sample-based adaptive sparse representation (AdaSR)", Pattern Recognition, vol. 44, no. 9, **(2011)**, pp. 2170–2183.
[23] T. B. Dinh and G. G. Medioni, "Co-training framework of generative and discriminative trackers with partial occlusion handling", Applications of Computer Vision (WACV), 2011 IEEE Workshop on, **(2011)**, pp. 642–649.
[24] H. Grabner, M. Grabner and H. Bischof, "Real-time tracking via online boosting", BMVC, vol. 1, no. 5, **(2006)**, pp. 47–56.

Authors



Hainan Zhao, received his M.S. degree in Computer Sciences from the Harbin Institute of Technology in 2009. Since 2010, he has been a Ph.D. degree candidate in Computer Sciences from Harbin Institute of Technology Shenzhen Graduate School. His research interests include computer vision, target tracking and pattern recognition.

Xuan Wang, received his M.S. and Ph.D. degrees in Computer Sciences from Harbin Institute of Technology in 1994 and 1997 respectively. He is a professor and Ph.D. supervisor in the Computer Application Research Center, Harbin Institute of Technology Shenzhen Graduate School. His main research interests include artificial intelligence, computer vision, computer network security and computational linguistics. He is a member of the IEEE.

Meng Liu received his M.S. degree in Computer Sciences from the School of Computer Science and Technology, Shandong University, in 2004. Currently, he is working toward the PhD degree in Computer Science at the Computer Application Research Center, Harbin Institute of Technology Shenzhen Graduate School and is a lecturer in the School of Mechanical, Electrical and Information Engineering, Shandong University at Weihai, China. His main research interests include computer vision, pattern recognition, and network and information security.

