Robust Object Tracking via Sparsity-based Collaborative Model

Wei Zhong
Dalian University of Technology
zhongwei@mail.dlut.edu.cn

Huchuan Lu
Dalian University of Technology
lhchuan@dlut.edu.cn

Ming-Hsuan Yang
University of California at Merced
mhyang@ucmerced.edu

Abstract

In this paper we propose a robust object tracking algorithm using a collaborative model. As the main challenge for object tracking is to account for drastic appearance change, we propose a robust appearance model that exploits both holistic templates and local representations. We develop a sparsity-based discriminative classifier (SDC) and a sparsity-based generative model (SGM). In the SDC module, we introduce an effective method to compute the confidence value that assigns more weights to the foreground than the background. In the SGM module, we propose a novel histogram-based method that takes the spatial information of each patch into consideration with an occlusion handing scheme. Furthermore, the update scheme considers both the latest observations and the original template, thereby enabling the tracker to deal with appearance change effectively and alleviate the drift problem. Numerous experiments on various challenging videos demonstrate that the proposed tracker performs favorably against several state-of-the-art algorithms.

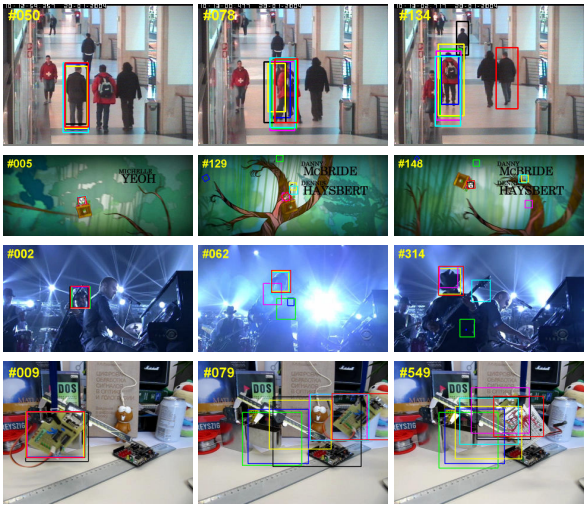


Figure 1. Tracking in challenging environments including heavy occlusions (*caviar*), rotation (*panda*), illumination change (*shaking*) and cluttered background (*board*). The results of the Frag [1], IVT [21], MIL [4], ℓ_1 [19], PN [12], VTD [13] tracking methods and our tracker are represented by cyan, blue, magenta, green, black, yellow and red rectangles, respectively.

1. Introduction

The goal of object tracking is to estimate the states of the target in image sequences. It plays a critical role in numerous vision applications such as motion analysis, activity recognition, video surveillance and traffic monitoring. While much progress has been made in recent years, it is still a challenging problem to develop a robust algorithm for complex and dynamic scenes due to large appearance change caused by varying illumination, camera motion, occlusions, pose variation and shape deformation (See Figure 1).

In a fixed frame, an appearance model is used to represent the object with proper features and verify predictions using object representations. In the successive frames, a motion model is applied to predict the likely state of an object (e.g., Kalman filter [6] and particle filter [20, 14]). In this paper, we focus on the appearance model since it is usually the most crucial component of any tracking algorithm.

Several factors need to be considered for an effective appearance model. First, an object can be represented by different features such as intensity [21], color [20], texture [3], superpixels [25], and Haar-like features [10, 11, 4, 12]. Meanwhile, the representation schemes can be based on holistic templates [6] or local histograms [1, 28]. In this work, we use intensity values for representation because of their simplicity and efficiency. Furthermore, our approach exploits both the strength of holistic templates to distinguish the target from the background, and the effectiveness of local patches in handling partial occlusion.

Second, a model needs to be developed to verify any state prediction, which can be either generative or discriminative. For generative methods, tracking is formulated as searching for the most similar region to the target object within a neighborhood [6, 1, 21, 19, 16, 15]. For discriminative methods, tracking is treated as a binary classification problem which aims at designing a classifier to distinguish

the target object from the background [2, 10, 3, 23, 11, 4, 12]. Furthermore, several algorithms have been proposed to exploit the advantages of both generative and discriminative models [31, 17, 22, 18, 7]. We develop a simple yet robust model that makes use of the generative model to account for appearance change and the discriminative classifier to effectively separate the foreground target from the background.

The third issue is concerned with online update schemes so that the tracker can adapt to appearance variations of the target object and the background. Numerous successful update approaches have been proposed [6, 10, 3, 21, 19]. However, straightforward and frequent updates of tracking results may gradually result in drifts due to accumulated errors, especially when the occlusion occurs. To address this problem, Babenko *et al*. [4] devise a strategy for choosing positive and negative samples during update and introduce multiple instance learning (MIL) to learn the true target object which is included in the positive bag. Kalal *et al*. [12] propose a bootstrapping classifier. They explore the structure of unlabeled data via positive and negative constraints which help to select potential samples for update. In order to capture appearance variations as well as reduce tracking drifts, we propose a method that takes occlusions into consideration for updating appearance model.

In this paper, we propose a robust object tracking algorithm with an effective and adaptive appearance model. We use intensity to generate holistic templates and local representations in each frame. Within our tracking scheme, the collaboration of generative models and discriminative classifiers contributes to a more flexible and robust likelihood function for particle filter. The appearance model is adaptively updated with the consideration of occlusions to account for variations and alleviate drifts. Numerous experiments on various challenging sequences show that the proposed algorithm performs favorably against the state-of-the-art methods.

2. Related Work

Sparse representation has recently been applied to vision problems [26], including image enhancement [29], object recognition [27], and visual tracking [19, 16, 15]. Mei and Ling [19] apply sparse representation to visual tracking and deal with occlusions via trivial templates. Despite of demonstrated success, there are still several issues to be addressed. First, the algorithm is able to deal with occlusion with ℓ_1 minimization formulation using trivial templates at the expense of high computational cost. Second, the trivial templates can be used to model any kind of image regions whether they are from the target objects or the background. Thus, the reconstruction errors of images from the occluded target and the background may be both small. As a result of generative formulation where the sample with minimal reconstruction error is regarded as the tracking result, ambiguities are likely to accumulate and cause tracking failure. Liu *et al*. [16] propose a method which selects a sparse and discriminative set of features to improve tracking efficiency and robustness. One potential problem with this approach is that the number of discriminative features is fixed, which may not be effective for tracking in dynamic and complex scenes. In [15], a tracking algorithm based on histograms of local sparse representation is proposed. The target object is located via mean-shift of voting maps constructed basing on reconstruction errors. In contrast to the histogram generation scheme in [15] that does not differentiate foreground and background patches, we propose a weighting method to ensure that the occluded patches are not used to account for appearance change of the target object, thereby resulting a more robust model. Furthermore, the average pooling method in [15] does not consider geometric information between patches while our method exploits the spatial information of local patches with histograms. In addition to model object appearance with local histograms, we also maintain a holistic template set that further helps identify the target object.

Occlusion is one of the most challenging problems in object tracking. Adam *et al*. [1] propose a fragments-based method to handle occlusions. The target is located by a voting map formed by comparing histograms of the candidate patches and the corresponding template patches. However, the template is not updated and sensitive to large appearance variations. Yang *et al*. [28] present the "bag of features" algorithm to visual tracking. Nevertheless, each local feature is assigned to the nearest codeword, which may result in loss of visual information [5] and ambiguity, especially when the features lie near the center of several codewords. This may lead to poor and unstable appearance representation of the target object and cause tracking failure. We develop an effective method which estimates and rejects possible occluded patches to improve robustness of appearance representation when occlusions occur. In addition, our tracker is adaptively updated with consideration of whether patches are occluded or not to better account for appearance change.

3. Proposed Algorithm

In this section, we present the proposed algorithm in details. We first discuss the motivation of this work. Next, we describe how the holistic and local visual information are exploited. The update scheme of our appearance method is then presented.

3.1. Problem Formulation

The representation schemes for object tracking mainly consist of holistic templates and local histograms. While most tracking algorithms use either holistic or local representations, our approach exploits the collaborative strength

of both schemes. Most tracking methods use rectangle to represent the tracking result, yet the pixels within the tracking rectangle are not all from foreground. As a result, the local representation-based classifier may be affected when updated with the background patches as positive samples. On the contrary, the holistic templates are often distinct to be foreground or background. Thus, the holistic templates are more suitable for discriminative models. Meanwhile, local representations are more amenable for generative models because of their flexibility. Therefore, we develop a collaborative model that integrates a discriminative classifier based on holistic templates and a generative model using local representations.

3.2. Sparsity-based Discriminative Classifier (SDC)

Motivated by the demonstrated success of sparse representation classifier [27], we propose our sparsity-based discriminative classifier for object tracking. For simplicity, we use the vector $\mathbf{x}$ to represent the gray-scale values of a target image.

3.2.1 Templates

The training image set is composed of N_p positive templates and N_n negative templates. Initially, we sample N_p images around the manually selected target location (e.g., within a radius of a few pixels). Then, the selected images are normalized to the same size (32×32 in our experiments) for efficiency. Each downsampled image is stacked to form the corresponding positive template vector. Similarly, the negative training set is composed of images further away from the marked location (e.g., within an annular region a few pixels away from the target object). In this way, the negative training set consists of both the background and images of parts of the target object. This facilities better object localization as samples containing only partial appearance of the target are treated as the negative samples and their confidence values are restricted to be small.

In each frame, we draw N candidates around the tracked result in the previous frame with a particle filter. To better track the target, we employ affine transformation [21] to model object motion. In addition, we assume that the affine parameters are independent and can be modeled with six scalar Gaussian distributions.

3.2.2 Feature Selection

The above-mentioned gray-scale feature space is rich yet redundant, from which determinative ones that distinguish foreground from background can be extracted. We select discriminative features by

$$\min_{\mathbf{s}} \left\| \mathbf{A}^{\top}\mathbf{s} - \mathbf{p} \right\|_2^2 + \lambda \|\mathbf{s}\|_1, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{K \times (N_p + N_n)}$ is composed of N_p positive templates $\mathbf{A}_+$ and N_n negative templates $\mathbf{A}_-$, and K is the feature dimension before feature selection. Each element of the vector $\mathbf{p} \in \mathbb{R}^{(N_p+N_n)\times 1}$ represents the property of each template in the training set $\mathbf{A}$, i.e., $+1$ for positive templates and -1 for negative templates. The solution of Eq. 1 is the sparse vector $\mathbf{s}$, whose nonzero elements correspond to discriminative features selected from the original K-dimension feature space. Note that the feature selection scheme adaptively chooses suitable number of discriminative features in the dynamic environment.

We project the original feature space to the selected feature space via a project matrix $\mathbf{S}$. It is formed by removing all-zero rows from a diagonal matrix $\mathbf{S}'$ where the elements are determined by

$$S'_{ii} = \begin{cases} 0, & s_i = 0 \\ 1, & \text{otherwise}, \end{cases} \tag{2}$$

where the diagonal element S'_{ii} is zero when s_i of $\mathbf{s}$ is zero. Both the training template set and the candidates sampled by a particle filter are projected to the selected and discriminative feature space. Thus, the training template set and candidates in the projected space are $\mathbf{A}' = \mathbf{SA}$ and $\mathbf{x}' = \mathbf{Sx}$.

3.2.3 Confidence Measure

The proposed SDC is developed based on the assumption that the target can be better represented by the linear combination of positive templates while the background can be better represented by the span of negative templates. Given the candidate, it is represented by the training template set with the coefficients α computed by

$$\min_{\alpha} \|\mathbf{x}' - \mathbf{A}'\alpha\|_2^2 + \lambda\|\alpha\|_1. \tag{3}$$

A candidate with smaller reconstruction error using the foreground template set indicates it is more likely to be a target object, and vice versa. Thus, we formulate the confidence value H_c of the candidate $\mathbf{x}$ by

$$H_c = \exp\left(-\left(\varepsilon_f - \varepsilon_b\right)/\sigma\right), \tag{4}$$

where $\varepsilon_f = \|\mathbf{x}' - \mathbf{A}'_+\alpha'_+\|_2^2$ is the reconstruction error of the candidate $\mathbf{x}$ with the foreground template set $\mathbf{A}_+$, and α_+ is the corresponding sparse coefficient vector. Similarly, $\varepsilon_b = \|\mathbf{x}' - \mathbf{A}'_-\alpha'_-\|_2^2$ is the reconstruction error of the candidate $\mathbf{x}$ using the background template set $\mathbf{A}_-$, and α_- is the related sparse coefficient vector. The variable σ is fixed to be a small constant that balances the weight of the discriminative classifier and the generative model presented in Section 3.3.

In [27], the authors employ the reconstruction error on the target (positive) templates. It is not quite appropriate for

tracking, since both the negative samples and the indistinguishable samples have large reconstruction errors on the target (positive) templates. Thus, it introduces ambiguity for the tracker. Our confidence measure exploits the distinction between the foreground and the background; its benefit is presented in Section 3.4.

3.3. Sparsity-based Generative Model (SGM)

Motivated by the success of sparse coding for image classification [30, 24, 9] as well as object tracking [15], we present a generative model for object representation that considers the location information of patches and takes occlusion into account.

3.3.1 Histogram Generation

For simplicity, we use the gray-scale features to represent the local information. We use overlapped sliding windows on the normalized images to obtain M patches and each patch is converted to a vector $\mathbf{y}_i \in \mathbb{R}^{G\times 1}$, where G denotes the size of the patch. The sparse coefficient vector β of each patch is computed by

$$\min_{\beta_i} \|\mathbf{y}_i - \mathbf{D}\beta_i\|_2^2 + \lambda\|\beta_i\|_1, \tag{5}$$

where the dictionary $\mathbf{D} \in \mathbb{R}^{G\times J}$ is generated from k-means cluster centers (J denotes the number of cluster centers) via the patches belonging to the labeled target object in the first frame and it consists of the most representative patterns of the target object.

In this work, the sparse coefficient vector $\beta_i \in \mathbb{R}^{J\times 1}$ of each patch is concatenated to form a histogram by

$$\rho = [\beta_1, \beta_2, \cdots, \beta_M]^\top, \tag{6}$$

where $\rho \in \mathbb{R}^{(J\times M)\times 1}$ is the proposed histogram for one candidate.

The average pooling scheme for histogram generation used in [15] is efficient, yet the strategy may miss the spatial information of each patch. For example, if we change the location of the left part and the right part of a human face image, the average pooling scheme neglects the exchange while our method will discover it.

3.3.2 Occlusion Handling

In order to deal with occlusions, we modify the constructed histogram to exclude the occluded patches when describing the target object. The patch with large reconstruction error is regarded as occlusion and the corresponding sparse coefficient vector is set to be zero. Thus, a weighted histogram is generated by

$$\varphi = \rho \odot \mathbf{o}, \tag{7}$$

where $\odot$ denotes the element-wise multiplication. Each element of $\mathbf{o}$ is an indicator of occlusion of the corresponding patch and is obtained by

$$o_i = \begin{cases} 1 & \varepsilon_i < \varepsilon_0 \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

where $\varepsilon_i = \|\mathbf{y}_i - \mathbf{D}\beta_i\|_2^2$ is the reconstruction error of patch $\mathbf{y}_i$, and ε_0 is a predefined threshold which determines the patch is occluded or not.

We thus have a sparsity-based histogram φ for each candidate. The proposed representation scheme takes spatial information of local patches and occlusion into account, thereby making it more effective and robust.

3.3.3 Similarity Function

We use the histogram intersection function to compute the similarity of histograms between the candidate and the template due to its effectiveness [9] by

$$L_c = \sum_{j=1}^{J\times M} \min\left(\varphi_c^j, \psi^j\right), \tag{9}$$

where φ_c and ψ are the histograms for the c-th candidate and the template.

The histogram of the template (denoted by ψ) is generated by Eqs. 5-7. The patches $\mathbf{y}$ in Eq. 5 are all from the first frame and the template histogram is computed only once for each image sequence. It is updated every several frames and the update scheme is presented in Section 3.5. The vector $\mathbf{o}$ in Eq. 8 reflects the occlusion condition of the corresponding candidate. The comparison between the candidate and the template should be carried out under the same occlusion condition, so the template and the c-th candidate share the same vector $\mathbf{o}_c$ in Eq. 7. For example, when the template is compared with the c-th candidate, the vector $\mathbf{o}$ of the template in Eq. 7 is set to $\mathbf{o}_c$.

3.4. Collaborative Model

We propose a collaborative model using SDC and SGM within the particle filter framework. In our tracking algorithm, the confidence value based on the holistic templates and the similarity function based on the local patches jointly contribute to an effective and robust description of the probability. The likelihood function of the c-th candidate is constructed by

$$\begin{aligned} p_c &= H_c L_c \\ &= \exp\left(-\left(\varepsilon_f - \varepsilon_b\right)/\sigma\right)\left(\sum_{j=1}^{J\times M} \min\left(\varphi_c^j, \psi^j\right)\right), \end{aligned} \tag{10}$$

and the tracking result is the candidate with the highest probability.

The multiplicative formula is more effective in our tracking scheme compared with the alternative additive scheme.

The confidence value H_c gives higher weights to the candidates considered as positive samples (i.e., ε_f smaller than ε_b) and penalizes the others. As a result, it can be considered as the weight of the local similarity function. Moreover, the confidence value of indistinguishable candidate (i.e., it can be equally constructed by positive and negative template sets when ε_f is almost equal to $\approx \varepsilon_b$) is equal to 1 and it has no effect on the likelihood function when multiplying with the local similarity function. Consequently, in the collaborative model, the SGM module plays a more important role in object tracking.

3.5. Update Scheme

Since the appearance of an object often changes significantly during the tracking process, the update scheme is important and necessary. We develop an update scheme in which the SDC and SGM are updated independently.

For the SDC model, we update the negative templates every several frames (5 in our experiments) from image regions away (e.g., more than 8 pixels) from the current tracking result. The positive templates remain the same in the entire sequence. As the SDC model aims at distinguishing the foreground from the background, it must make sure that the positive templates and the negative templates are all correct and distinct. In this way, the SDC model is adaptive and discriminative.

For the SGM model, the dictionary $\mathbf{D}$ is fixed for the same sequence. Therefore, the dictionary is not deteriorated by the update of tracking failures or occlusions. In order to capture the new appearance and recover the object from occlusions, the template histogram is updated by

$$\psi_n = \mu\psi_f + (1-\mu)\,\psi_l \quad \text{if } O_n < O_0, \tag{11}$$

where the new histogram ψ_n is composed of the histogram ψ_f at the first frame and the histogram ψ_l last stored according to the weights assigned by the constant μ. The variable O_n denotes the occlusion condition of the tracking result in the new frame. It is computed by the corresponding occlusion indication vector $\mathbf{o}_n$ (by Eq. 8) using

$$O_n = \sum\nolimits_{i=1}^{J\times M} \left(1 - o_n^i\right). \tag{12}$$

The update is performed as long as the occlusion condition O_n in this frame is smaller than a predefined constant O_0. The update scheme preserves the first template which is usually correct and takes the newly arrived template into account.

4. Experimental Results

In order to evaluate the performance of our tracker, we conduct experiments on ten challenging image sequences. These sequences cover most challenging situations in object tracking: heavy occlusion, motion blur, in-plane and out-of-plane rotation, large illumination change, scale variation and complex background (See Figure 3). For comparison, we run six state-of-the-art algorithms with the same initial position of the target. These algorithms are the Frag tracking [1], IVT tracking [21], MIL tracking [4], ℓ_1 tracking [19], PN tracking [12] and VTD tracking [13] methods. We present some representative results in this section. All the MATLAB source codes and datasets are available on our web sites (http://ice.dlut.edu.cn/lu/publications.html, http://faculty.ucmerced.edu/mhyang/pubs.html).

The parameters are presented as follows. Note that they are fixed for all sequences. The numbers of positive templates N_p and negative templates N_n are 50 and 200 respectively. The variable λ in Eq. 1 is fixed to be 0.001. The variable λ in Eqs. 3 and 5 is fixed to be 0.01. The row number G and column number J of dictionary $\mathbf{D}$ in Eq. 5 are 36 and 50. The threshold ε_0 in Eq. 8 is 0.04. The update rate μ is set to be 0.95. The threshold O_0 in Eq. 11 is 0.8.

Table 1. Average center location error (in pixel). The best and second best results are shown in red and blue fonts.

	Frag	IVT	MIL	ℓ_1	PN	VTD	Our
animal	92.1	127.5	66.5	15.3	–	**12.0**	**10.8**
board	**45.4**	165.4	66.7	184.0	90.1	105.0	**12.7**
car11	64.0	**2.2**	43.5	33.3	25.2	27.1	**1.8**
caviar	116.1	66.0	100.2	65.7	**44.5**	58.3	**2.7**
faceocc2	15.5	**10.3**	14.1	11.2	18.6	10.5	**4.8**
girl	**18.1**	48.5	32.3	62.5	23.2	21.5	**9.8**
jumping	58.5	36.9	9.9	12.5	**3.6**	63.0	**3.8**
shaking	52.8	152.7	11.2	118.7	–	**6.1**	**9.4**
singer1	22.1	8.5	15.2	4.6	32.7	**4.1**	**3.8**
panda	**90.1**	169.8	103.4	94.0	–	94.8	**2.5**

Table 2. Average overlap rate based on [8]. The best and second best results are shown in red and blue fonts.

	Frag	IVT	MIL	ℓ_1	PN	VTD	Our
animal	0.07	0.21	0.21	0.53	0.41	**0.57**	**0.59**
board	**0.65**	0.14	0.46	0.12	0.34	0.32	**0.78**
car11	0.08	**0.80**	0.17	0.43	0.37	0.43	**0.79**
caviar	0.13	0.14	0.13	0.13	**0.16**	0.15	**0.85**
faceocc2	0.60	0.58	0.61	**0.67**	0.49	0.59	**0.81**
girl	**0.68**	0.42	0.51	0.32	0.57	0.51	**0.69**
jumping	0.13	0.28	0.52	0.55	**0.69**	0.07	**0.73**
shaking	0.24	0.02	0.65	0.03	0.12	**0.73**	**0.67**
singer1	0.34	0.66	0.33	0.70	0.41	**0.79**	**0.85**
panda	0.23	0.15	0.35	0.16	**0.60**	0.36	**0.69**

4.1. Quantitative Comparison

We evaluate the above-mentioned algorithms using the center location error as well as the overlapping rate [8], and the results are shown in Table 1 and Table 2. Figure 2 shows the center location errors of the evaluated algorithms on all test sequences. Overall, the proposed tracker performs well against the other state-of-the-art algorithms.

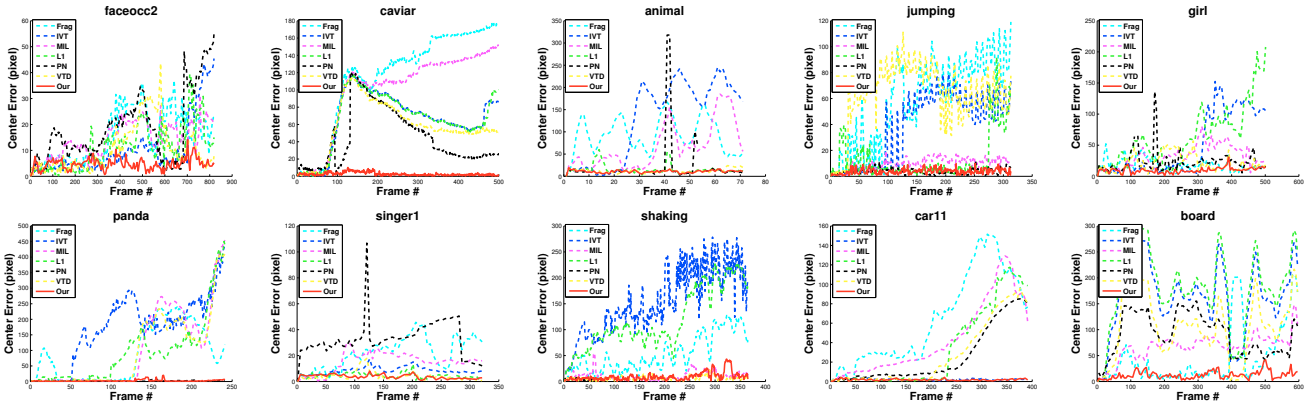


Figure 2. Quantitative evaluation in terms of center location error (in pixel). The proposed algorithm is compared with six state-of-the-art methods on ten challenging image sequences.

4.2. Qualitative Comparison

Heavy occlusion: Occlusion is one of the most general yet crucial problems in object tracking. In fact, several trackers including the FragTrack method [1], the MIL tracking algorithm [4], the ℓ_1 tracking method [19] and our tracker are developed to solve this problem. In contrast, the IVT tracking method [21], the PN tracking method [12] and the VTD tracking system [13] are less effective in handling occlusions as shown in Figure 3(a), especially at frames 175, 497, 819 of the *faceocc2* sequence. In our SGM module, we estimate the possible occluded patches and develop a robust histogram which only compares the patches that are not occluded. Thus, the occlusion handling scheme effectively alleviates the affect of occlusions. Aside from tracking a target object under occlusion, our method updates appearance change correctly especially when heavy occlusions occur. In addition, our tracker is able to deal with in-plane rotation when the target is occluded at frame 497, owing to the appearance model we employ. Our tracker can accurately locate the target object at frame 819 as our generated histogram takes the spatial information of local patches into consideration.

In the *caviar* sequence, the target is occluded by two people at times and one of them is similar in color and shape to the target. The other trackers all fail before frame 134 due to heavy occlusion (Figure 3(a)). Furthermore, for most template-based trackers, simple update with occluded portion often leads to drifts (frame 442 of Figure 3(a)). In contrast, our tracker achieves stable performance in the entire sequence when there is a large scale change with heavy occlusion. This can be attributed to our SGM model that reduces the effect of occlusions and only compares the foreground with the stored histograms. Besides, our update scheme doesn't introduce heavy occlusions which may lead to drift problem.

Motion blur: Fast motion of the target object or the camera leads to blurred image appearance which is difficult to account for in object tracking. Figure 3(b) presents the tracking results on the *animal* sequence in which the appearance of the target object is almost indistinguishable due to the motion blur. Most tracking algorithms fail to follow the target right at the beginning of this sequence. At frame 42, the PN tracking method [12] mistakenly locates a similar object instead of the correct target. The reason is that the true target is blurred and it is difficult for the detector of P-N [12] to distinguish it from the background. The proposed algorithm well handles the situation with similar objects as the SDC module selects the discriminative features to better separate the target from the background. By updating the negative templates online, the proposed algorithm successfully tracks the target object throughout the sequence.

The appearance change caused by motion blur in the *jumping* sequence is drastic that the Frag [1] and VTD [13] methods fail before frame 31. The IVT [21] method is able to track the target in some frames (e.g., frame 100) but fails when the motion blur occurs (e.g., frame 238). Our tracker successfully keeps track of the target object with small errors. The main reason is that we use the SDC module which separates the foreground from the background. Meanwhile, the confidence measure by Eq. 4 assigns smaller weights to the candidate of background. Thus, the tracking result will not drift to the background.

Rotation: The *girl* sequence in Figure 3(c) consists of both in-plane and out-of-plane rotations. The PN tracking method [12] and the VTD tracking method [13] fail when the girl rotates her head. Compared with other algorithms, our tracker is more robust and accurate as seen from frame 312 and frame 430. In our tracking scheme, the background candidates are assigned quite small weights according to Eq. 4. Therefore, the tracking result will not shift to the background when the girl rotates (e.g., frame 111 and frame 312).

The target object in the *panda* sequence experiences more and larger in-plane rotations. As seen from frame 53, the IVT method [21] fails due to occlusion and fast movement. Most trackers drift after the target undergoes large rotations (e.g., frame 154) whereas our method performs well throughout this sequence. As the other trackers often account for object motion with translational or similarity transforms, they are not able to deal with complex movements. In addition, the use of local histograms helps in accounting for appearance change due to complex motion. Furthermore, the target object in the *panda* sequence also undergoes occlusions as shown in frame 53 and frame 214. The PN tracking method [12] fails to detect occlusions and track the target object after frame 214 while our tracker still performs well.

Illumination change: Figure 3(d) presents the tracking results on sequences with dramatic illumination changes. In the *singer1* sequence, the stage light changes drastically seen from frame 121 and frame 321. The PN tracking method [12] is not able to detect and track the target object (e.g., frame 121). On the other hand, our tracker accurately locates the target object even when there is a large scale change at frame 321. In the *shaking* sequence, the target object undergoes large appearance variation due to drastic illumination change and unpredictable motion. Our SDC module introduces the backgrounds and the images with parts of the target as negative templates so the confidence values of these candidates calculated by Eq .4 are small. Thus, the tracking result is accurately located on the true target without much offset.

For the *car11* sequence, there is low contrast between the foreground and the background (frame 284) as well as illumination change. The FragTrack method [1] fails at the beginning (at frame 19) because it only uses the local information and does not maintain a holistic representation of the target. The IVT tracking method [21] achieves good results in this sequence. It can be attributed to the fact that subspace learning method is robust to illumination changes. In our SDC module, we select several discriminative features which can better separate the target from the background. Thus, our tracker performs well in spite of the low contrast between the foreground and the background.

Complex background: The *board* sequence is challenging as the background is cluttered and the target object experiences out-of-plane rotations as seen from Figure 3(e). In frame 55, most trackers fail as holistic representations inevitably include background pixels that may be considered as part of foreground object through straightforward update schemes. Using fixed templates, the FragTrack method [1] is able to track the target as long as there is no drastic appearance change (e.g., frame 55 and frame 183), but fails when the target moves quickly or rotates (e.g., frame 78, frame 395 and frame 528). Our tracker performs well in this sequence as the target can be differentiated from the cluttered background with the use of our SDC module. In addition, the update scheme uses the newly arrived negative templates that facilitate separation of the foreground object and the background.

5. Conclusion

In this paper, we propose and demonstrate an effective and robust tracking method based on the collaboration of generative and discriminative modules. In our tracker, holistic templates are incorporated to construct a discriminative classifier that can effectively deal with cluttered and complex background. Local representations are adopted to form a robust histogram that considers the spatial information among local patches with an occlusion handling module, which enables our tracker to better handle heavy occlusion. The contributions of these holistic discriminative and local generative modules are integrated in a unified manner. Moreover, the online update scheme reduces drifts and enhances the proposed method to adaptively account for appearance change in dynamic scenes. Quantitative and qualitative comparisons with six state-of-the-art algorithms on ten challenging image sequences demonstrate the robustness of our tracker.

Acknowledgements

W. Zhong and H. Lu are supported by the National Natural Science Foundation of China #61071209. M.-H. Yang is supported by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, 2006.

[2] S. Avidan. Support vector tracking. *PAMI*, 26(8):1064–1072, 2004.

[3] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261–271, 2007.

[4] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.

[5] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

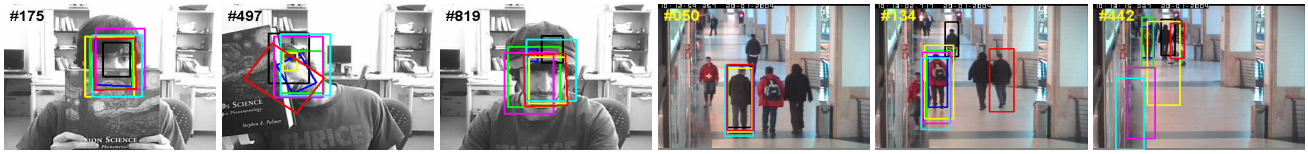
[6] D. Comaniciu, V. R. Member, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–575, 2003.

[7] T. B. Dinh and G. G. Medioni. Co-training framework of generative and discriminative trackers with partial occlusion handling. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pages 642–649, 2011.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results, 2010.

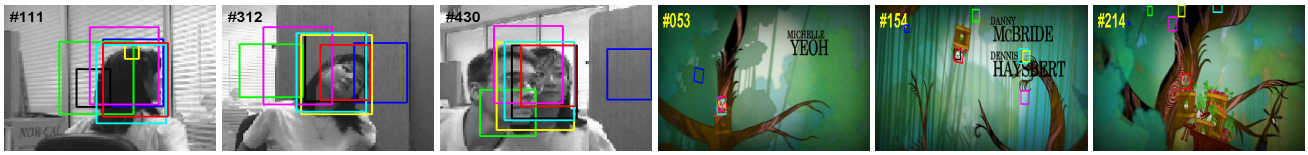
[9] S. Gao, I. W.-H. Tsang, L.-T. Chia, and P. Zhao. Local features are not lonely - laplacian sparse coding for image classification. In *CVPR*, 2010.

[10] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, 2006.



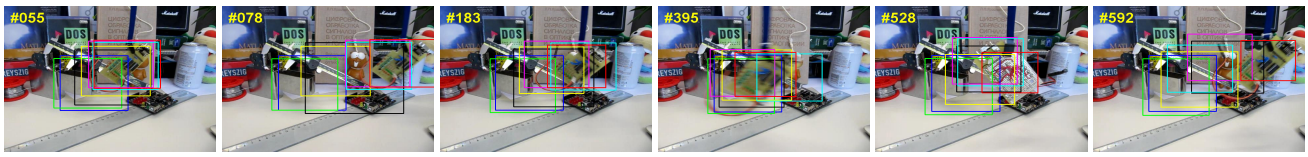
(a) Tracking results on sequence *faceocc2* and *caviar* with severe occlusions.

(b) Tracking results on sequence *animal* and *jumping* with motion blur.



(c) Tracking results on sequence *girl* and *panda* with in-plane and out-of-plane rotations.

(d) Tracking results on sequence *singer1*, *shaking* and *car11* with dramatic illumination change.



(e) Tracking results on sequence *board* with complex background.

Figure 3. Sample tracking results of evaluated algorithms on ten challenging image sequences.

[11] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008.

[12] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010.

[13] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, 2010.

[14] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans. *PAMI*, 30(10):1728–1740, 2008.

[15] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *CVPR*, 2011.

[16] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. Robust and fast collaborative tracking with two stage sparse optimization. In *ECCV*, 2010.

[17] R. Liu, J. Cheng, and H. Lu. A robust boosting tracker with minimum error bound in a co-training framework. In *ICCV*, 2009.

[18] H. Lu, Q. Zhou, D. Wang, and X. Ruan. A co-training framework for visual tracking with multiple instance learning. In *FG*, 2011.

[19] X. Mei and H. Ling. Robust visual tracking using ℓ_1 minimization. In *ICCV*, 2009.

[20] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV*, 2002.

[21] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[22] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel robust online simple tracking. In *CVPR*, 2010.

[23] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In *ICCV*, 2007.

[24] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[25] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011.

[26] J. Wright, Y. Ma, J. Maral, G. Sapiro, T. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[27] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 31(2):210–227, 2009.

[28] F. Yang, H. Lu, and Y.-W. Chen. Bag of features tracking. In *ICPR*, 2010.

[29] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *TIP*, 19(11):2861–2873, 2010.

[30] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[31] Q. Yu, T. B. Dinh, and G. G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *ECCV*, 2008.