Partial Occlusion Handling for Visual Tracking via Robust Part Matching

Tianzhu Zhang*, Kui Jia*, Changsheng Xu, Yi Ma, Narendra Ahuja
Institute of Automation, Chinese Academy of Sciences, P. R. China
Advanced Digital Sciences Center of Illinois, Singapore
School of Information Science and Technology, ShanghaiTech University
University of Illinois at Urbana-Champaign, Urbana, IL USA

Abstract

Part-based visual tracking is advantageous due to its robustness against partial occlusion. However, how to effectively exploit the confidence scores of individual parts to construct a robust tracker is still a challenging problem. In this paper, we address this problem by simultaneously matching parts in each of multiple frames, which is realized by a locality-constrained low-rank sparse learning method that establishes multi-frame part correspondences through optimization of partial permutation matrices. The proposed part matching tracker (PMT) has a number of attractive properties. (1) It exploits the spatial-temporal locality-constrained property for robust part matching. (2) It matches local parts from multiple frames jointly by considering their low-rank and sparse structure information, which can effectively handle part appearance variations due to occlusion or noise. (3) The proposed PMT model has the inbuilt mechanism of leveraging multi-mode target templates, so that the dilemma of template updating when encountering occlusion in tracking can be better handled. This contrasts with existing methods that only do part matching between a pair of frames. We evaluate PMT and compare with 10 popular state-of-the-art methods on challenging benchmarks. Experimental results show that PMT consistently outperforms these existing trackers.

1. Introduction

Visual tracking is one of the fundamental problems in computer vision. Its real-world applications range from video surveillance, autonomous vehicles, intelligent traffic control, human-computer interaction, etc. However, visual tracking is challenging due to significant object appearance variations caused by illumination change, occlusion, sensory noise, fast/abrupt object motion, and also cluttered background. Over the past years, tremendous efforts in visual tracking has been made to overcome these challenges, yielding a steady performance improvement. However, most of existing methods [4, 14, 20, 24, 32] focus on modeling the holistic appearance of the target. As a result, the tracking is prone to fail especially in the presence of partial occlusion, as shown in Figure 1(a).
To design a robust tracking algorithm in spite of partial occlusion, researchers have developed sophisticated appearance models through statistical analysis \[15\], robust statistics \[1\] \[10\], model analysis \[12\], learning occlusion with likelihoods \[21\], and sparse representation \[24\] \[35\] \[34\] \[5\]. Among them, methods part-wisely modeling object appearance \[1\] \[13\] \[9\] \[27\] \[18\] \[29\] \[5\] become more popular partially because of their favorable property of robustness against partial occlusion. Indeed, when there exists partial occlusion, some parts of the object remain visible which provide reliable cues for tracking. Most of these methods can be viewed as tracking by part-based object matching over time in a video sequence. However, they have the following drawbacks. (1) Most of them track each part independently and ignore the collaborations among parts. Parts in one frame should be jointly matched to the corresponding parts in the consecutive frame. (2) Most of them establish part matchings between a pair of frames, while ignoring the very same object target appearing in other adjacent or history frames, which may provide additional constraints helpful for part matching. In addition, these methods usually propagate the part matching result in the present frame to subsequent frames, which may accumulate matching errors and are consequently prone to losing track. As a result, these existing part matching based tracking methods are still less reliable when more complicated factors of appearance variations appear in the video.

It is thus desirable that a globally consistent part matching among multiple frames can be established in visual tracking. To achieve this goal, we propose in this paper a new tracking algorithm based on the following observations: (1) In a short duration, if appearance of individual parts of object remains unchanged, their intensities in video frames should be similar. Representing appearance of an object part as a vector, the matrix formed by the vectors of the corresponding parts in multiple frames of the short duration should be low-rank, ideally rank-one. We are thus motivated to use the low-rank assumption as a criteria for part matching. (2) If there exist object appearance variations in images due to occlusion, object pose change, or illumination change, the low-rank assumption in (1) cannot be fully satisfied. To alleviate their negative effects, we may decompose out these appearance variations in images as sparse errors so that the low-rank assumption still applies. (3) Matching of individual parts from multiple adjacent frames should satisfy the locality-constrained property. In spatial domain, parts in one frame should be jointly matched to the parts in other frames. In temporal domain, the matched parts in adjacent frames should satisfy the constant-velocity motion constraint. (4) A dictionary of multi-mode target templates should be maintained and progressively updated to model the target appearance variations, which is critical to correct the track after occlusion. (5) Part matching across multiple frames is more robust than that between a pair of frames, as it can leverage additional constraints from other frames that contain the very same target.

Motivated by these observations, we propose in this paper a novel method, termed Part Matching Tracker (PMT), for robust visual tracking. PMT realizes part matching among multiple frames by optimizing a partial permutation matrix for each frame, using locality-constrained low-rank and sparsity of matched parts as criteria. Compared with existing part based visual tracking methods, our proposed PMT has three major contributions. (1) PMT has the spatial and temporal locality-constrained property, which enables our tracking of local parts to satisfy the constant-velocity motion constraint. (2) Part tracking using PMT is based on rank and sparsity optimization, which is potentially effective to model part appearance variations due to occlusion, illumination change, or target pose change over time. (3) Our tracker operates in a batch mode, in which multi-mode target templates and frames to be tracked are simultaneously taken into account to determine a global matching of corresponding parts. Even if occlusion happens, the error would not be propagated to the subsequent frames, and the track can be inferred from the observations before and after. Therefore, our tracker effectively cope with partial occlusion as shown in Figure \[16\]. We intensively compare with 10 popular state-of-the-art methods on challenging benchmarks. Experimental results show that PMT consistently outperform these existing trackers.

2. Related Work

In general, visual tracking methods can be categorized as either generative or discriminative. Generative methods use appearance models to represent the target object and search for the most similar image regions to the generative model. Popular generative trackers include eigentracker \[7\], incremental tracker \[20\] \[23\], sparse trackers \[24\] \[34\] \[5\], visual tracking decomposition \[23\], and so on. A drawback of these methods is that they are not designed to distinguish between target and background patches, and are prone to drift. Discriminative methods formulate object tracking as a binary classification, which aims to find the target location that can best distinguish the target from the background. Popular discriminative methods include on-line boosting \[14\], ensemble tracking \[3\], online multiple instance learning \[4\], tracking-learning-detection \[20\], struck tracker \[16\], compressive tracker \[31\], etc. Most of these methods, however, delineate the entire tracked target by a single regular bounding box, which renders them sensitive to partial occlusion and damages tracking performance.

Part based visual tracking draws more recent attention. In \[18\] \[27\], multiple people tracking is achieved by part based model motivated by its successful application in human detection \[11\]. In \[17\], each part is tracked inde-
pendently, and the results are treated as multiple measurements [28]. Tracking is then achieved by identifying inconsistent measurements. The Frag tracker [1] models object appearance with histogram of local parts and combines votes of matching local patches. However, this template is not updated and therefore it is not expected to handle appearance changes. Nejhum et al. [25] model object shape in terms of a small number of rectangular blocks. The drawback is that it requires manual initialization of part locations carefully. Godec et al. [13] extend the hough forest to the online domain and integrate the voting method for tracking, regardless of the parts’ spatial-temporal correlations.

Our formulation of leveraging low-rank sparse property for optimization of partial permutation matrices is similar to [19], which addresses feature matching across a set of images. However, [19] cannot address the visual tracking problem due to its ignorance of the fundamental spatial-temporal locality-constrained property. Furthermore, we formulate each corresponding part of the target object as a low-rank matrix, while all features to be matched are formulated into one low-rank matrix in [19]. Consequently, when using techniques in [19], matching of different parts may interfere with each other and may not be able to well address the partial occlusion problem in tracking.

3. Our Proposed Part Matching Tracker

In this section, we give details of our proposed PMT that is based on a locality-constrained low-rank sparse learning method to optimize partial permutation matrices for the part correspondence problem among multiple frames.

3.1. Problem Setup

A typical setting of the tracking problem is that an object identified, either manually or automatically, in the first frame of a video sequence is tracked in the subsequent frames by estimating its bounding boxes as it moves. As discussed in previous sections, tracking methods that delineate the tracked object by a single regular bounding box will render them sensitive to partial occlusion and significantly impact their tracking performance. To address this problem, we attempt to adopt a part-based model to describe the target. The advantage of this model comes from the observation that under partial occlusion conditions, some parts of the object remain visible and distinguishable and can provide reliable cues for tracking. Therefore, if we can infer occlusion information from the confidence scores of individual parts, we can consequently utilize only the parts with high confidence to estimate the position of target over time.

To obtain the confidence scores of individual parts, we can adopt part matching methods. In the situation of visual tracking, a moving object appears in multiple frames of a video sequence. A straightforward approach is to locally build part correspondences between pairs of frames. However, pair-wise matching cannot leverage additional constraints from other frames that also contain the very same target. It may thus be less robust to noise and occlusion of parts. In addition, multi-mode target templates should be maintained to model the variations of target appearance over the history frames, which makes it possible to infer the tracker even after occlusion. Therefore, parts should be matched with a more global and consistent property across the sequence and in the target templates, in order to achieve robustness against occlusion. As discussed in Section [1] parts in a video volume have the spatial-temporal locality-constrained property, and appearance of the same local parts across frames have the low-rank sparse property. To exploit these properties, we propose a locality-constrained low-rank sparse learning method for robust part matching among multiple frames, which include both the multi-model target templates and frames to be tracked.

3.2. Problem Formulation

We sample $K_1$ target templates at and around the position of object in the first frame, as did in [4][24][14]. These target templates are of equal size. They will be progressively updated to incorporate variations of object appearance due to changes in illumination, viewpoint, etc. Target appearance remains the same only for a certain period of time. Eventually the templates are no longer accurate representations of the object appearance. A fixed target template is prone to the tracking drift problem, since it is insufficient to handle changes in appearance. Conversely, if the target templates are updated too often, irrelevant variations will be more possible to be introduced into the templates, causing tracking drift. In this work, we use the target template update scheme as in [24], where the tracking result is added to the template set if none of the templates are similar to the tracking result. For the template, their parts can be extracted by dividing each template into regular grids. Tracking of the object in the incoming frames is realized by matching its candidate parts to those in the target templates. Candidate parts in the incoming frames are simply sampled by particle filtering [2] at and around the parts of the previous tracking results by considering their recursive weights.

For the $K_1$ target templates, we extract $n_k$ parts from each of them, $k = 1, \ldots, K_1$. These parts are all from the object target. We denote $K_2$ as the number of incoming frames to be tracked, and $K = K_1 + K_2$. For each of the incoming frames, we also sample $n_k$ parts, $k = 1, \ldots, K_2$. The sampled parts from incoming frames are possibly background patches. For simplicity of notation, we use the same $n_k$ to index parts of target templates and those of incoming frames, i.e., $n_k$ for $k = 1, \ldots, K$. We denote the feature vectors associated with individual parts of any $k \in \{1, \ldots, K\}$ as $F^k = [f^k_1, \ldots, f^k_{n_k}] \in \mathbb{R}^{d \times n_k}$, and assume that these feature vectors in $\{F^k\}_{k=1}^K$ are not corre-
sponded with respect to each other. Our interest is to find \( n \leq n_k, \forall k \in \{1, \ldots, K\} \), intrinsic parts from each target template or incoming frame, and establish their correspondences. Because we model their correspondences based on multiple frames and target templates (\( K \)), our tracker has the globally consistent property. The correspondences on multiple frames and target templates (\( \{ \text{template or incoming frame} \} \)), corresponded with respect to each other. Our interest is to find the others are zeros. The deficient, ideally rank one. Here, \( \{ P^k \}_{k=1}^K \) is defined as follows:

\[
P^k = \{ P^k | P^k \in \{0, 1\}^{n_k \times n}, 1_{n_k}^T P^k = 1_n^T, P^k 1_n \leq 1_{n_k} \},
\]

where \( \{0, 1\}^{n_k \times n} \) denotes a \( n_k \times n \) matrix whose elements are either 0 or 1 and \( 1_{n_k} \) denotes a column vector of all 1 of length \( n \). The term \( 1_{n_k}^T P^k = 1_n^T \) in Eq. (1) shows that the \( i^{th} \) part, \( i = 1, \ldots, n \), is corresponded to only one of all the \( n_k \) sampled parts. The term \( P^k 1_n \leq 1_{n_k} \) constrains that each sampled part is corresponded to at most one of all the \( n \) parts. Motion smoothness in visual tracking implies that the sampled parts of the \( i^{th} \) part in next frame must be associated to part \( i \) in current frame. So each of the rows of \( P^k \) that correspond to samples for part \( i \) in frame \( k \) must have exactly one element equal to 1. This constraint is written by defining a \( n \times n \) matrix \( A^k \) whose \( i^{th} \) row flags the samples for part \( i \) in frame \( k \), and requiring that \( A^k P^k 1_n = 1_n \).

\[
A^k = \begin{bmatrix} 1 & 0 \cdots & 0 \\ 0 & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}, \quad A^k \in \mathbb{R}^{n \times n}
\]

Here, \( A^k \in \mathbb{R}^{n \times n} \) and its \( i^{th} \) row \( A^k_i \) is defined as follows: the elements from the \( m_i^k + m_{i-1}^k + 1 \) to \( m_i^k + m_{i-1}^k + 1 \) are 1, and the others are zeros. The \( m_i^k \) is the number of sampled parts for the \( i^{th} \) part of the \( k^{th} \) image, and \( n_k = m_1^k + \ldots + m_n^k \).

\[
P^k = \{ P^k | P^k \in \{0, 1\}^{n_k \times n}, 1_{n_k}^T P^k = 1_n^T, P^k 1_n \leq 1_{n_k}, A^k P^k 1_n = 1_n \},
\]

As a result, Eq. (2) can satisfy the spatial-temporal locality-constrained property of parts among multiple frames.

Features of the correspondents in different target templates or incoming frames should be linearly correlated. Let \( \{ P^k \}_{k=1}^K \) in Eq. (2) be the optimized partial permutation matrices such that parts are re-ordered and well corresponded, we thus have \( D_i = [F^1 p_i^1, \ldots, F^K p_i^K] \in \mathbb{R}^{d \times K}, i = 1, \ldots, n \), which stacks the features of the \( i^{th} \) part from target templates or incoming frames as a matrix, and \( D_i \) is rank deficient, ideally rank one. Here, \( p_i^k \) is the \( i^{th} \) column of \( P^k \) and \( p_i^k = P^k e_i \), where \( e_i \) denotes a unit column vector with all entries set to 0 except the \( k^{th} \) one, which is set to 1. Therefore, the problem of optimizing partial permutation matrices \( \{ P^k \}_{k=1}^K \) can be formulated as the following rank minimization problem:

\[
\min_{\{ P^k \}_{k=1}^K, \{ L_i \}_{i=1}^n} \sum_{i=1}^n \text{rank}(L_i)
\]

\text{s.t.} \quad D_i = L_i, \quad i = 1, \ldots, n. \quad (3)

In many visual tracking scenarios, target objects are often contaminated by noise, illumination change, object pose change, or partial occlusion. As a result, the parts characterizing the same local appearance information of object in different frames could vary. Thus the low-rank assumption used in [3] is likely to be violated. To improve the robustness, we introduce a sparse error term into [3] to model the noise of the data matrix \( D_i \), where we assume these errors are sparse and only appear in a small fraction of \( D_i \). Therefore, in the presence of noise or occlusion, the problem (3) can be refined as follows:

\[
\min_{\{ P^k \}_{k=1}^K, \{ L_i, E_i \}_{i=1}^n} \sum_{i=1}^n \text{rank}(L_i) + \lambda \| E_i \|_0
\]

\text{s.t.} \quad D_i = L_i + E_i, \quad i = 1, \ldots, n. \quad (4)

where \( \| \cdot \|_0 \) is \( \ell_0 \)-norm counting the number of nonzero entries, and \( \lambda > 0 \) is a parameter controlling the trade-off between rank property of \( L_i \) and sparsity of \( E_i \). As a result, the part matching problem in Eq (4) guarantees the low-rank sparse property.

4. Optimization

It is not tractable to solve the problem (4) due to the following aspects: (1) The two terms \( \text{rank}(\cdot) \) and \( \| \cdot \|_0 \) are non-convex, discrete-valued functions; (2) The entries of \( \{ P^k \}_{k=1}^K \) are constrained to be binary. To make it tractable, we first make use of the convex surrogates \( \| \cdot \|_s \) and \( \| \cdot \|_1 \) to replace \( \text{rank}(\cdot) \) and \( \| \cdot \|_0 \), respectively. Here, \( \| \cdot \|_s \) denotes nuclear norm (sum of the singular values) and \( \| \cdot \|_1 \) is \( \ell_1 \)-norm. Applying the relaxation strategy to (4) yields

\[
\min_{\{ P^k \}_{k=1}^K, \{ L_i, E_i \}_{i=1}^n} \sum_{i=1}^n \| L_i \|_s + \lambda \| E_i \|_1
\]

\text{s.t.} \quad D_i = L_i + E_i, \quad i = 1, \ldots, n. \quad (5)

To simplify the subsequent notations, we change the variables and rewrite the formulation (5) as follows:

\[
\min_{\{ \theta^k \}_{k=1}^K, \{ L_i, E_i \}_{i=1}^n} \sum_{i=1}^n \| L_i \|_s + \lambda \| E_i \|_1
\]

\text{s.t.} \quad D = L + E, \quad \theta^k \in \{0, 1\}^{n_k \times n}, k \in \{1, \ldots, K\}, \quad Q^k \theta^k = 1_n, H^k \theta^k \leq 1_{n_k}, S^k \theta^k = 1_{n_k}, \quad (6)

where \( \theta^k = \text{vec}(P^k), vee(P^k) \) is the vectorization of the matrix \( P^k \), \( \theta^k = \text{vec}(P^k) \) is the vectorization of the matrix \( P^k \), \( G^k = I_n \otimes F^k \in \mathbb{R}^{d_n \times n_k}, D = [I_{n_1} + E_1]^T, \ldots, [I_{n_n} + E_n]^T \in \mathbb{R}^{d_n \times n_k}, D = [G^1 \theta^1, \ldots, G^K \theta^K], Q^k = I_n \otimes 1_{n_k} \in \mathbb{R}^{n \times n_k}, H^k = 1_n^T \otimes I_{n_k} \in \mathbb{R}^{n \times n_k}, S^k = 1_n^T \otimes A^k \in \mathbb{R}^{d_n \times n_k}, \infty \) is the Kronecker product, and \( I_n \) (or \( I_{n_k} \)) is the identity matrix of size \( n \times n \) (or \( n_k \times n_k \)). Here, we have used the fact vec(\( XYZ \)) = vec(\( Y \) \( X \)). The (6) involves jointly optimizing a set of \( K \) partial permutation matrices, exact solution of which is NP-hard. To get an approximate solution, we use the fast first-order Alternative Direction Method of Multiplier (ADMM) [3]. The general ADMM decomposes a global
problem into local subproblems that can be readily solved. For \(K\) subproblems, ADMM decomposes optimization of \(L_i, E_i, \{P_i^k\}_{k=1}^K\) into subproblems that update \(L_i, E_i\), and each of \(\{P_i^k\}_{k=1}^K\), respectively. The augmented Lagrangian function of the optimization problem (6) can be written as:

\[
\mathcal{L} \left( \{L_i, E_i\}_{i=1}^n, \{\theta^k\}_{k=1}^K, Y, u \right) = \sum_{i=1}^n \left( \|L_i\|_s + \lambda \|E_i\|_1 \right) + \langle Y, D - L - E \rangle + \frac{u}{2} \|D - L - E\|_F^2 \tag{7}
\]

\[
\Rightarrow \min_{\{L_i, E_i\}_{i=1}^n, \{\theta^k\}_{k=1}^K} \mathcal{L} \left( \{L_i, E_i\}_{i=1}^n, \{\theta^k\}_{k=1}^K, Y, u \right)
\]

where \(Y \in \mathbb{R}^{dn \times K}\) is a matrix of Lagrange multipliers, \(u\) is a positive scalar, \((\cdot, \cdot)\) denotes the inner product, and \(\|\cdot\|_F\) denotes the Frobenius norm. The ADMM algorithm algorithmically updates one of the matrices \(L_i, E_i, \{P_i^k\}_{k=1}^K\), and the Lagrange multiplier \(Y\) by minimizing (7), while keeping the others fixed to their most recent values. Consequently, we obtain three update steps corresponding to the three sets of variables. The details are the following:

**Step 1: Update \(L_i\) and \(E_i\), \(\forall i \in \{1, \ldots, n\}\) (with others fixed):** The minimization problem (7) w.r.t. \(\{L_i, E_i\}_{i=1}^n\) can be decomposed into \(n\) independent subproblems (each of them is corresponding to one part). The \(i^{th}\) subproblem to update \(L_i\) and \(E_i\) can be equivalently rewritten:

\[
\{L_i, E_i\} = \arg \min_{L_i, E_i} \left\{ \|L_i\|_s + \lambda \|E_i\|_1 + \langle Y, D_i - L_i - E_i \rangle + \frac{u}{2} \|D_i - L_i - E_i\|_F^2 \right\} \tag{8}
\]

Then, the solution of (8) can be obtained by solving the optimization problems in Eq (9) and Eq (10), respectively.

\[
L_i = \arg \min_{L_i} \frac{1}{u} \|L_i\|_s + \frac{1}{2} \|L_i - D_i + E_i - Y_i/u\|_F^2 \tag{9}
\]

\[
E_i = \arg \min_{E_i} \frac{\lambda}{u} \|E_i\|_1 + \frac{1}{2} \|E_i - D_i + L_i - Y_i/u\|_F^2 \tag{10}
\]

**Step 2: Update \(\theta^k\), \(\forall k \in \{1, \ldots, K\}\) (with others fixed):** The minimization problem (7) with respect to \(\{\theta^1, \ldots, \theta^K\}\) can be decomposed into \(K\) independent subproblems, each of which corresponds to \(\{\theta^k\}\) and can be equivalently formulated as the following integer constrained convex quadratic programming (QP) problem:

\[
\theta^k = \arg \min_{\theta^k} \frac{1}{2} \theta^k \mathbf{G} \mathbf{G}^\top \theta^k + e_k^{\top} \left( \frac{Y}{u} - D \right) \mathbf{G}^\top \theta^k \tag{11}
\]

\[
\text{s.t. } \theta^k \in \{0, 1\}^{n \times n}, Q^n \mathbf{G} \theta^k = 1_n, H^n \mathbf{G} \theta^k \leq 1_{n_k}, S^n \mathbf{G} \theta^k = 1^n \tag{12}
\]

where \(D = L + E\). This is a NP-hard problem. However, as proved for a very similar problem in [19],

\[
\| \mathbf{G} \mathbf{G}^\top \theta^k \|_2^2 = \sum_{i=1}^n \| \mathbf{F}_i^k p_i^k \|^2_2
\]

is a constant value if the features are normalized. Therefore, the quadratic term of problem (11) can be removed to get the linear programming problem (12). By relaxing the binary constraint to a real value between 0 and 1, the problem (12) can be exactly solved by a standard LP solver.

\[
\theta^k = \arg \min_{\theta^k} e_k^{\top} \left( \frac{Y}{u} - D \right) \mathbf{G}^\top \theta^k \tag{13}
\]

**Step 3: Update Multiplier \(Y\):** We update the Lagrange multipliers in Eq (13), where \(\rho > 1\).

\[
Y = Y + u(D - L - E); \quad u = pu \tag{13}
\]

5. Experimental Results

**Datasets:** We evaluate tracking performance on 16 publicly available video sequences, which are captured in different scenarios and contain challenging appearance variations due to occlusion, object pose and scale changes, illumination change, and abrupt motion.

**Implementation Details:** In all experiments, the number of target templates is set to \(K_1 = 5\) as in most of existing trackers. We set \(K_2 = 3\), and \(\lambda = 1\) (in Eq (4)). We adopt Geometric Blur [6] as the feature to characterize each part. As a trade-off between effectiveness and speed, \(n\) and \(n_k\) are set to 6, 100, respectively. Here, we employ a simple heuristic to determine the number of parts \(n\) within the tracking object as in [30] - we divide the object into six parts of either \(3 \times 2\) or \(2 \times 3\) depending on its aspect ratio.

**Baselines:** Our PMT tracker is analyzed and compared with 10 state-of-the-art tracking methods, FRAGT (fragment-based tracker [1]), VTD (visual tracking decomposition [22]), \(L_i T\) (\(l_1\) minimization tracker [24]), IVT (incremental subspace visual tracker [25]), MIL (multiple instance learning tracker [4]), LRST (low-rank sparse tracker [33]), TLD (tracking-learning-detection [28]), CT (compressive tracking [31]), Struck (structured output tracker [16]), and OAB (online AdaBoost [14]). We implement these trackers using publicly available source codes or binaries provided by the authors. The parameters of these trackers are adjusted to show the best tracking performance. For fair comparisons, the same initializations are set to all methods. The supplementary material contains result videos.

**Evaluation Metrics:** For quantitative comparison, two popular evaluation metrics are used. The first metric is the
and Struck algorithms throughout the entire sequence despite large illumination changes. The “david”, “singer”, and “trellis” contains significant illumination changes and pose variations. On the “trellis” sequence, Frag, and VT fail to drift away from the target after frame 172 because of the changing lighting conditions. Due to the combination of lighting and head pose changes, IVT, Frag, and CT fail to track the target after the 367th frame. Both our tracker and Struck successfully track the target across the entire video sequence, although our tracker locates the head more accurately. The sequences “girl” and “skating” contain abrupt motion, pose change, and partial occlusion. On the “girl” sequence, the proposed tracker is capable of tracking the target for the entire sequence. Other trackers experience drift at different time. The “soccer” sequence contains abrupt motion and background clutter. Compared with other trackers, PMT achieves the better results and can track the target object despite scale and pose changes as well as occlusion by confetti at most of the frames. In contrast, other methods (IVT, L1, OAB, MIL, and Frag) fail to track the target reliably. The “biker” sequence contains scenes with abrupt motion and large pose change. Nevertheless, our PMT performs well throughout the entire sequence with more stable tracking results.

5.2. Discussion and Analysis

We present more illustrative tracking examples in this section to demonstrate the effectiveness of PMT for robust visual tracking. In particular, Figure 4 demonstrates a process of partial occlusion, where for each of the three sets of incoming frames to be tracked (around frames 248, 266, 295), we only show one of the target templates on the left and one of the tracked frames on the right, due to space limit. When the partial occlusion starts at the incoming frame 248, PMT still matches its parts to those of the target template shown on the top-left image. However, due to partial occlusion, the three parts on the left of the face in the top-right image rank higher in terms of matching confidence. Appearance of the partially occluded face is very different from the one in the target templates, PMT thus updates the target template as shown in the middle-left image. After update-

<table>
<thead>
<tr>
<th>Video</th>
<th>PMT</th>
<th>L1</th>
<th>MIL</th>
<th>OAB</th>
<th>VTD</th>
<th>IVT</th>
<th>Frag</th>
<th>Struck</th>
<th>TLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>car4</td>
<td>14.4</td>
<td>16.0</td>
<td>15.3</td>
<td>15.0</td>
<td>14.3</td>
<td>15.1</td>
<td>15.0</td>
<td>15.8</td>
<td>14.2</td>
</tr>
<tr>
<td>soccer</td>
<td>15.6</td>
<td>15.8</td>
<td>15.0</td>
<td>14.8</td>
<td>15.4</td>
<td>15.3</td>
<td>15.2</td>
<td>15.8</td>
<td>14.2</td>
</tr>
<tr>
<td>skating</td>
<td>14.9</td>
<td>15.0</td>
<td>14.3</td>
<td>14.1</td>
<td>14.6</td>
<td>14.4</td>
<td>14.3</td>
<td>14.9</td>
<td>14.2</td>
</tr>
<tr>
<td>singer</td>
<td>3.7</td>
<td>3.8</td>
<td>3.9</td>
<td>4.0</td>
<td>3.7</td>
<td>3.8</td>
<td>3.7</td>
<td>3.9</td>
<td>3.8</td>
</tr>
<tr>
<td>girl</td>
<td>3.5</td>
<td>3.7</td>
<td>3.8</td>
<td>3.9</td>
<td>3.7</td>
<td>3.8</td>
<td>3.7</td>
<td>3.9</td>
<td>3.8</td>
</tr>
<tr>
<td>face1</td>
<td>6.2</td>
<td>6.4</td>
<td>6.2</td>
<td>6.4</td>
<td>6.2</td>
<td>6.4</td>
<td>6.2</td>
<td>6.4</td>
<td>6.2</td>
</tr>
<tr>
<td>face2</td>
<td>7.6</td>
<td>7.8</td>
<td>7.6</td>
<td>7.8</td>
<td>7.6</td>
<td>7.8</td>
<td>7.6</td>
<td>7.8</td>
<td>7.6</td>
</tr>
<tr>
<td>david</td>
<td>10.6</td>
<td>10.8</td>
<td>10.7</td>
<td>10.7</td>
<td>10.7</td>
<td>10.7</td>
<td>10.7</td>
<td>10.7</td>
<td>10.7</td>
</tr>
<tr>
<td>coke1</td>
<td>7.8</td>
<td>8.0</td>
<td>8.1</td>
<td>8.1</td>
<td>8.1</td>
<td>8.1</td>
<td>8.1</td>
<td>8.1</td>
<td>8.1</td>
</tr>
<tr>
<td>car1</td>
<td>3.4</td>
<td>3.6</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>bike1</td>
<td>10.4</td>
<td>10.6</td>
<td>10.6</td>
<td>10.6</td>
<td>10.6</td>
<td>10.6</td>
<td>10.6</td>
<td>10.6</td>
<td>10.6</td>
</tr>
<tr>
<td>soccer</td>
<td>15.8</td>
<td>16.0</td>
<td>15.8</td>
<td>15.8</td>
<td>15.8</td>
<td>15.8</td>
<td>15.8</td>
<td>15.8</td>
<td>15.8</td>
</tr>
<tr>
<td>olsr1</td>
<td>3.8</td>
<td>3.9</td>
<td>3.9</td>
<td>3.9</td>
<td>3.9</td>
<td>3.9</td>
<td>3.9</td>
<td>3.9</td>
<td>3.9</td>
</tr>
<tr>
<td>olsr2</td>
<td>2.5</td>
<td>2.7</td>
<td>2.7</td>
<td>2.7</td>
<td>2.7</td>
<td>2.7</td>
<td>2.7</td>
<td>2.7</td>
<td>2.7</td>
</tr>
</tbody>
</table>
Figure 3. Tracking results of 11 trackers (denoted in different colors) on 16 video sequences. Frame numbers are overlayed in yellow. See text for details. Results best viewed on high-resolution displays.

Figure 2. Quantitative comparison of the 11 trackers with the center location error on the 4 video sequences. The most of the six parts at frame 266 match well with the updated target template, as shown by the confidence scores in the middle-right image. This update of target template is important when partial occlusion remains for a longer duration of time, otherwise matching may fail and tracking drifts. When the face re-appears at frame 295, PMT matches its parts to those of earlier target templates that contain no occlusion, as shown in the bottom image of Figure 4. The tracking process continues successfully, which shows the effectiveness of our proposed PMT.

Figure 4. Illustration of PMT’s robustness against partial occlusion. The numbers of “1” to “6” index different parts of the face. “1” ranks highest and “6” ranks lowest in terms of confidence score of part matching. More explanations are in Section 5.2.
Our PMT seems less prone to noise, part matching error, or partial occlusion. This is further demonstrated in Figure 5, where when some of the parts cannot be matched to those in the target templates, other parts of the object are less influenced and their matchings still make tracking successful. This is consistent with confidence scores of the matching of different parts shown in the right of Figure 5.

6. Conclusion

In this paper, we proposed a locality-constrained low-rank sparse learning method to effectively optimize optimal partial permutation matrices for the part correspondence among multiple frames for visual tracking. By using the three properties (locality-constrained property, low-rank sparse property, and globally consistent property), our tracker is robust for partial occlusion. We extensively analyze the performance of our tracker on challenging real-world video sequences and show it outperforms 10 state-of-the-art tracking methods.

Acknowledgment

This study is supported by the research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore’s Agency for Science, Technology and Research (A*STAR). C. Xu is supported by 973 Program Project No. 2012CB316304 and NSFC 61225009.

References


Figure 5. Our PMT can track well even when there are errors of part matching due to occlusion, illumination changes, etc.