Detection-free Bayesian Multi-object Tracking via Spatio-Temporal Video Bundles Grouping

Technical Report, November 2013

Yongyi Lu, Liang Lin, Yuanlu Xu, Zefeng Lai

Abstract

This paper presents a conceptually simple but effective approach to track multi-object in videos without requiring elaborate supervision (i.e. training object detectors or templates offline). Our framework performs a bi-layer inference of spatio-temporal grouping to exploit rich appearance and motion information in the observed sequence. First, we generate a robust middle-level video representation based on clustered point tracks, namely video bundles. Each bundle encapsulates a chunk of point tracks satisfying both spatial proximity and temporal coherency. Taking the video bundles as vertices, we build a spatio-temporal graph that incorporates both the competitive and compatible relations among vertices. The multi-object tracking can be then phrased as a graph partition problem under the Bayesian framework, and we solve it by developing a robust belief propagation (RBP) algorithm. This algorithm improves the traditional belief propagation method by allowing a converged solution to be reconfigured during optimization, so that the inference can be re-activated once it gets stuck in local minima and thus conduct more reliable results. In the experiments, we demonstrate the superior performances of our approach on the challenging benchmarks compared with state-of-the-arts.

I. Introduction

Many efforts have been taken for object tracking in computer vision, as well as impressive progresses obtained recently. Many state-of-the-art tracking methods are detection-guided [17], [9], which usually rely on pre-trained or online-maintained object detectors to predict object states during tracking. Despite acknowledged successes, these methods, however, would have problems on the following scenarios: (i) Frequent partial occlusion and object deformation lower the precision of detector. (ii) The detection responses are possibly inconsistent in time, resulting in the risk of tracking drift. (iii) For some objects with large intra-class variance (e.g. sports players), the cost of training reliable detectors is expensive.

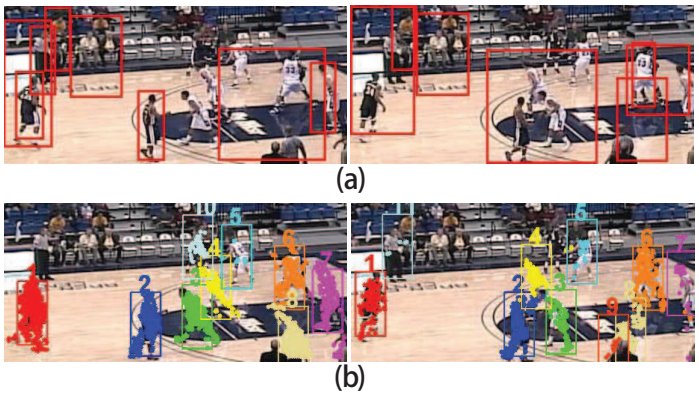

Fig. 1. An example of detection-free human tracking on the complex scenario. (a) shows two frames from a sport video, where the human detections (denoted by the red boxes) are unreliable due to the appearance variations and occlusions. (b) shows the tracking results generated by our approach. The object IDs (denoted by the numbers) are retained well during tracking and the human silhouettes are basically preserved.

In this paper, we present a detection-free tracking framework that parses object trajectories in the observed video sequence via spatio-temporal grouping without adopting object detectors. Our framework infers multiple-object tracking with two stages: (i) Extract a batch of **video bundles** by encapsulating dense point tracks to compose object trajectories, and (ii) Associate identities of the bundles for trajectory parsing by a robust belief propagation (**RBP**) algorithm. The inference is conducted based on a set of deferred observations (e.g. the entire video or a period of frames). Fig. 1 demonstrates the advantages of our approach: Some detection-guided methods may not work as the human detections are unreliable, while the satisfied results are produced by our approach.

The video bundle can be regarded as a intermediate-level video representation of object trajectory, just like the superpixel in image segmentation. A video bundle comprises of clustered dense point tracks that can be in different lengths over consecutive frames, and one trajectory may include a batch of bundles in the video. The video bundle advances in the following aspects, compared with traditional region-based representations [29], [24]. First, the point tracks are clustered in terms of satisfying both spatial proximity and temporal coherency, so that the bundles are more robust against noises and object conglutinations. Second, the bundles, in the form of spatio-temporal volumes, effectively reduce the complexity during inference, i.e. without the need of extracting frame-based correspondences in each step of tracking. Moreover, object silhouettes can be basically preserved by the bundles and we thus obtain fine-grained object trajectories from the video.

Taking video bundles as graph vertices, we link each vertex to its neighbors with an edge in 3D

coordinate to construct an spatio-temporal graph. The edge can be either positive or negative, indicating the two vertices either cooperatively or conflictingly belong to the same trajectory. The negative (competitive) relations serve as important complements to the positive (compatible) ones, both of which should be satisfied with probabilities during inference. We assign an edge to be positive or negative by examining the moving directions of the two video bundles. Specifically, if two video bundles have significantly different moving directions, they are less likely to belong to the same object trajectory. Then we pose the multi-object tracking as a graph partition task under the Bayesian framework.

For the inference of graph partition, we present the RBP algorithm adapting the real challenges during tracking. In computer vision, Belief Propagation (BP) algorithm and its extensions are widely employed for graph-based inference [22], [4], providing a general way to assign labels to graph vertices. However, these algorithms sometimes suffer from getting stuck in unsatisfied local minima. This problem might be more serious in tracking, as we need to simultaneously capture spatial and temporal information of objects and scenes. In this work, we improve the traditional BP algorithm by allowing the solution to be reconfigured during optimization. We verify the converged solution using the constraint of global trajectory consistency, and re-activate the inference (i.e. to jump out from the local minima), by operating on current solution. In brief, our algorithm iterates with the two steps: (i) Searching for a solution of graph partition by passing messages and updating beliefs. (ii) Reconfiguring the graph partition by realizing the merge-and-split operators on graph vertices, once the solution violates the constraint.

The rest of this paper is organized as follows. Section 2 reviews the related work. We introduce the representations of our approach in Section 3, and discuss the formulation and inference procedure in Section 4 and Section 5, respectively. The experiments and comparisons are presented in Section 6, and finally comes the conclusion in Section 7.

II. Related Work

Object tracking, in general, is a joint task of object segmentation and temporal correspondence over video sequences. Under the circumstances that reliable object models (e.g. detectors or templates) are invalid or limited, one can solve object tracking as spatio-temporal pixel grouping (or association) in the bottom-up manners [24], [15]. These methods shared some techniques with those for video segmentation [26]. For example, S. Wang et al [24] utilized structure information captured by superpixels, and proposed to distinguish the targets and background with the mid-level cues. In [17], a trajectory graph and a detection tracklet graph were constructed to encode grouping affinities in space and associations across, respectively. Basharat et al. [3] proposed to construct motion segments based on the spatial and

temporal analysis of interest point correspondence. Other works [26], [15], [11] utilized dense point tracks to represent object trajectories and demonstrated very good results for capturing motion discontinuities across object boundaries.

The key to success in multi-object tracking is the inference algorithm of data association. The main challenges lie in various ambiguities during optimization caused by abrupt object motion, non-rigid object structures, and changing appearance patterns of both objects and scenes. Previous approaches usually dealt with these problems by exploiting appearance and/or motion cues from different perspectives, in which the graphical representations were widely adopted [19], [10], [23], [30], [2], [1]. Exemplar inference algorithms included linear programming [8], dynamic programming [9], joint probabilistic data-association filter [20], [23]. Liu et al. [28] performed the stochastic cluster sampling for parsing trajectory in a spatio-temporal graph, but the algorithm is computationally expensive. Our framework is partially motivated by these methods, and advances them in two aspects. First, the representation of video bundles tightly integrates the spatial and temporal information to reduce the ambiguities of multi-object tacking. Second, the proposed RBP algorithm is very robust and fast to conduct reliable results by incorporating both the competitive and compatible relations among moving objects.

In the experiments, we compare with both the detection-free tracking frameworks [15], [26], [16] and the detection-based methods [17], [9] on public benchmarks, and our algorithm performs favorably against all competing methods.

III. Representation

We first introduce the video bundle representation and the problem formulation under the spatio-temporal graph.

A. Video Bundle

We first define a point track τ_i to be a sequence of points:

$$\tau_i = \{\, p_{i,k} : \; k \in [t_{i,b},\, t_{i,e}] \,\}, \tag{1}$$

where $p_{i,k}$ indicates the spatial coordinate of τ_i at frame k, $t_{i,b}$ and $t_{i,e}$ the birth and death time of τ_i, respectively.

We obtain point tracks from deferred video sequences by a recently proposed method [26], which tracks points densely using large displacement optical flow (LDOF). LDOF produces spatially-denser tracks than conventional track-clustering methods such as KLT [12], resulting in denser coverage of the moving targets.

The obtained track set contains point tracks generated from both foreground and background. Concentrating on the moving targets in foreground, we remove tracks belonging to background using the motion saliency proposed in [15]. The non-salient ones are treated as background tracks and discarded without further consideration.

We further group point tracks based on an affinity matrix A. Each element A_{ij} in the affinity matrix A measures the similarity between two tracks τ_i and τ_j. We define the similarity following two aspects: (i) geometric location and (ii) velocity,

$$A_{ij} \propto \exp\{-\mathcal{D}_{tw}(\tau_i, \tau_j)\} \cdot \exp\{-\frac{1}{|O_{ij}|} \sum_{k \in O_{ij}} \|v_{i,k} - v_{j,k}\|^2\}, \tag{2}$$

where O_{ij} denotes the frames τ_i and τ_j have temporal overlap, $v_{i,k} = p_{i,k+3} - p_{i,k}$ indicates the velocity for the k-th temporally-overlapped point of τ_i aggregated over 3 frames. $\mathcal{D}_{tw}(\cdot)$ is the dynamic time warping (DTW) distance [6] which measures the aligned geometric distance between two tracks. Given two tracks τ_i and τ_j, DTW seeks the warping path γ with minimum cost to align all points in each track

$$\mathcal{D}_{tw}(\tau_i, \tau_j) \;=\; \min \; \frac{1}{|\gamma|} \sqrt{\sum_{\gamma_k} \left\| p_{i,k_i} - p_{j,k_j} \right\|^2}\,, \tag{3}$$

where $\gamma_k = (p_{i,k_i}, p_{j,k_j})$ denotes the k-th aligned point pair in the warping path, $|\gamma|$ the total number of aligned point pairs. For detailed explanations, see [6].

Given the affinity matrix A, we adopt spectral clustering to group point tracks, a common technique utilized in image and video segmentation [13], [26]. This embeds the tracks into a lower dimensional subspace by finding the K smallest eigenvectors via eigen-decomposition, and we further group the embedded tracks using K-means. We call the obtained cluster a video bundle and denote it as $b_i = (\bar{\tau}_i, \bar{v}_i, \{\tau_j\})$, where $\bar{\tau}_i$ and $\bar{v}_i$ denote the cluster center and the mean velocity of b_i, respectively. $\bar{\tau}_i$ and $\bar{v}_i$ are computed by taking average over all point tracks belonging to b_i

$$\begin{aligned} \bar{\tau}_i &= \left\{ \bar{p}_{i,k} :\; \bar{p}_{i,k} = \frac{1}{|b_i|} \sum\nolimits_{j=1}^{|b_i|} p_{j,k},\; k \in \left[\min_j t_{j,b},\, \max_j t_{j,d} \right] \right\}, \\ \bar{v}_i &= \left\{ \bar{v}_{i,k} :\; \bar{v}_{i,k} = \frac{1}{|b_i|} \sum\nolimits_{j=1}^{|b_i|} v_{j,k},\; k \in \left[\min_j t_{j,b},\, \max_j t_{j,d} - 3 \right] \right\}, \end{aligned} \tag{4}$$

where $|b_i|$ denotes the number of tracks within b_i. The obtained video bundles, as shown in Fig. 2(a), provide robust and compact descriptions for moving objects.

B. Spatio-Temporal Graph

The objective of multi-target tracking is to identify the trajectory for each object in the video. Given the set of video bundles $B = \{b\}$, we define the solution W as

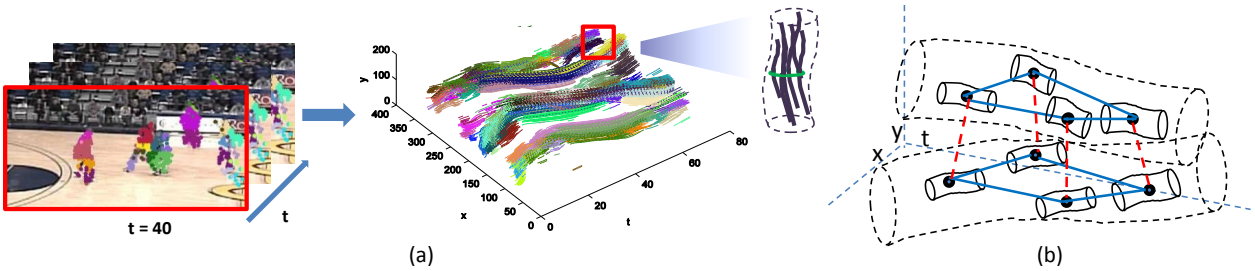

Fig. 2. An illustration of our representations. (a) shows the video bundle generated by point tracks exhibiting high affinities. In (b), the spatio-temporal graph is constructed by taking the bundles as vertices, and the blue and red edges indicate compatible and competitive relations between vertices, respectively. Best view in color image.

$$W = \{\, \Gamma_n = \{\, b_i,\ i = 1, 2, \ldots, |\Gamma_n| \,\},\ n = 1, 2, \ldots, N,\ b_i \in B \,\}, \tag{5}$$

where Γ_n denotes the trajectory for the n-th object and N the total number of objects in video. We constrain each trajectory encapsulating at least one bundle and each observed bundle belonging to one and only one trajectory. Thus, we formulate the problem of multi-object tracking as a graph partition task, i.e. grouping bundles into object trajectories.

We introduce a spatio-temporal graph $G = <B, E>$ to describe the relations among bundles. Each bundle $b_i \in B$ is taken as a graph vertex and each edge $e_{ij} = <b_i, b_j> \in E$ describes the relation between two adjacent (neighboring) bundles b_i and b_j. Two bundles b_i and b_j are regarded as neighbors $b_i \in \mathbb{N}(b_j)$ if they have temporal overlap $O_{ij} \neq 0$. We further develop two kinds of edges: negative edges E^- and positive edges E^+ to describe the competitive and compatible relations among them. Two neighboring bundles with significantly different motion directions yield a negative edge and otherwise a positive edge; namely,

$$E = E^- \cup E^+ = \{e_{ij} : \bar{v}_i \cdot \bar{v}_j < 0\} \cup \{e_{ij} : \bar{v}_i \cdot \bar{v}_j \geq 0\}. \tag{6}$$

For notation simplicity, we drop the notation of the edge index ij in the following discussion.

Negative edges penalize two bundles moving in the opposite direction being coupled together, i.e. these two bundles should belong to two different objects. We define a negative edge probability $\rho^-(b_i, b_j)$ to represent the extent of two bundles repulsing each other:

$$\rho_{ij}^- \propto \exp\{\, \bar{v}_i \cdot \bar{v}_j \,\}. \tag{7}$$

In other words, two bundles are less likely to belong to the same object if their motions are obviously different.

Positive edges encourage two bundles sharing similar statistics to be assigned with the same label. A positive edge probability ρ_{ij}^{+} is defined following two aspects: (i) the geometric distance and (ii) the temporal consistency,

$$\rho_{ij}^{+} \propto \exp\{-\mathcal{D}_{tw}(\bar{\tau}_i, \bar{\tau}_j)\} \cdot \exp\{-\mathcal{D}_{tc}(\bar{\tau}_i, \bar{\tau}_j)\}, \tag{8}$$

where $\bar{\tau}_i$ and $\bar{\tau}_j$ are the cluster centers for b_i and b_j, as defined in Equ.(4). $\mathcal{D}_{tw}(\cdot)$ is the DTW distance defined in Equ.(3). $\mathcal{D}_{tc}(\cdot)$ explores the motion context to provide a complementary cue for identifying the degree of attractiveness of two bundles. For example, in complex scenarios where numerous people move in diverse directions, bundles from different people may not have distinct motion difference. To overcome this problem, we proposed to measure their accumulated temporal consistency. Specifically, we connect one line segment between two bundles and accumulate the derivatives of optical flow along the line, and define,

$$\mathcal{D}_{tc}(\tau_i, \tau_j) = \frac{1}{|O_{ij}|} \sum_{k \in O_{ij}} \sum_{p \in \overline{p_{i,k}\, p_{j,k}}} \nabla F_k(p), \tag{9}$$

where $\overline{p_{i,k}\ p_{j,k}}$ denotes the line segment between points $p_{i,k}$ and $p_{j,k}$, and $\nabla F_k = (\frac{\partial F_k}{\partial x}, \frac{\partial F_k}{\partial y})$ the derivative of optical flow at the k-th frame.

An illustration of the spatio-temporal graph representation is shown in Fig. 2(b). Note we only focus on a few bundles in the red bounding box for clear specification.

IV. Bayesian Formulation

We solve W by maximizing a posterior probability under the Bayesian framework:

$$W^* = \arg\max_{W} P(W|B) \propto \arg\max_{W} P(B|W)P(W). \tag{10}$$

Likelihood $P(B|W)$ measures how well the observed data (video bundle) satisfies a certain object trajectory. Assuming the likelihood of each bundle is calculated independently given the partition, then $P(B|W)$ can be factorized into

$$P(B|W) = \prod_{\Gamma_n \in W} \prod_{b_i \in \Gamma_n} P(b_i|\Gamma_n). \tag{11}$$

Existing related methods [27], [26], [15] usually defined the likelihood model by maintaining a template for each specific object obtained by online learning. In this work, we define $P(b_i|\Gamma_n)$ by matching the observed bundle with the long-term trajectory it belongs to. By the very nature of a deformable agent, the long trajectories propagate their affinities further and are more stable than local representations. Specifically, we collect representative superpixels occupied by the trajectory Γ_n according to their sizes and temporal coherence (i.e. the locations over frames). These superpixels are then clustered into K

groups according to their appearance information. In the implementation, we concatenate a HSV color histogram of 40 bins as the appearance descriptor H, and we pool the feature over all superpixels in each cluster. Let H_k be the appearance feature of the k-th cluster. We define the likelihood model by matching b_i with its most similar cluster, as

$$P(b_i|\Gamma_n) \propto \exp\left\{-\min_k \mathcal{D}_{kl}(\, H(b_i) \parallel H_k(\Gamma_n)\,)\right\}, \tag{12}$$

where $\mathcal{D}_{kl}(\cdot)$ represents the the symmetric Kullback-Leibler Distance (KL).

Prior $P(W)$ imposes constraints on object trajectories and their interactions. We decompose such constraints into pairwise potentials between video bundles within each trajectory, that is

$$\begin{aligned} P(W) &= \prod_{\Gamma_n \in W} P(\Gamma_n) \prod_{\Gamma_n, \Gamma_m \in W} P(\Gamma_n, \Gamma_m) \\ &= \prod_{b_i, b_j \in \Gamma_n, e \in E^-} (1-\rho_{ij}^-) \prod_{b_i, b_j \in \Gamma_n, e \in E^+} \rho_{ij}^+ \prod_{b_i \in \Gamma_n, b_j \in \Gamma_m, e \in E^-} \rho_{ij}^- \prod_{b_i \in \Gamma_n, b_j \in \Gamma_m, e \in E^+} (1-\rho_{ij}^+), \end{aligned} \tag{13}$$

where ρ_{ij}^- and ρ_{ij}^+ are the negative and positive edge probability defined in Equ.(7-8).

V. Inference Algorithm

Given the spatio-temporal graph representation, inferring graph partition for W^* is a non-shallow problem, not only because the convexity guaranty of probability distribution $P(W|B)$ does not hold, but also due to the unknown number of targets. In our framework, we present a Robust Belief Propagation (RBP) algorithm to efficiently search for solutions while improving the effectiveness by introducing the trajectory-based global constraints. We pose the graph partition as the task of assigning labels to graph nodes. Let $\mathcal{L}$ be a set of labels, i.e., $\mathcal{L} = \{l_i = n, n = 1, 2, \ldots, N, b_i \in B\}$. A labeling l assigns a label l_i to each bundle $b_i \in B$.

A. Initialization

In some traditional belief propagation inferences, the algorithms initialize beliefs for nodes according to their unary likelihoods. In this work, we improve the initialization by further imposing pairwise similarities between vertices. Specifically, we can find a set of bundles as the representatives $\theta_i = \{\tilde{b}_k\}$ by comparing the similarity measure of bundles defined in Equ.(8). We first initialize the beliefs of representative bundles $\tilde{b}_k$ as 1 for their belonging labels and 0 for other labels. For each of the rest bundles b_j, we then compute the mean positive edge probabilities between b_j and the representatives for each label $\tilde{b}_k \in \theta_{l_i}$. Its belief is thus initialized as a distribution proportional to the mean edge probabilities for each labels and we put it into the set with the maximum belief. This process serves as a rough partition on the bundle set and gives more intuitive hints on the tracking solution.

B. Priority-based Message Passing

Given the spatio-temporal graph $G = <B, E>$, belief propagation iterates on exchanging messages between nodes and updating node beliefs. In the following discussion, we denote the message passed from b_i to b_j and the belief for a node b_j at the t-th iteration as $\Phi^t_{i\to j}(l_i)$ and $\Psi^t_j(l_j)$, respectively.

We adopt the mechanism of priority-based message passing proposed by [4] to suppress the ambiguous information passed between nodes. The intuition of this mechanism is to disambiguate the labels of nodes in virtue of the strength of their neighbors. The ambiguity of a node b_i at the t-th iteration is defined as the entropy of its current belief

$$\zeta^t(b_i) \;=\; -\sum_{l_i=1}^{N} \Psi^t_i(l_i)\log(\Psi^t_i(l_i)). \tag{14}$$

Nodes with less ambiguity are scheduled to transmit their messages with higher priority. Furthermore, to prevent propagating confusing information between nodes, a node only computes the messages passed from its less ambiguous neighbors. At the t-th iteration, the message passed from node b_i to node b_j is defined as

$$\Phi^t_{i\to j}(l_i) \propto \sum_{l_i=1}^{N} P(l_i, l_j)P(b_i|l_i) \prod_{b_k\in\mathbb{N}_<(b_i)} \Phi^{t-1}_{k\to i}(l_i), \tag{15}$$

where $\mathbb{N}_<(b_i)$ denotes the less ambiguous neighbors of b_i, i.e. $\mathbb{N}_<(b_i) = \{b_j : \zeta^t(b_j) < \zeta^t(b_i), b_j \in \mathbb{N}(b_i)\}$. Note an implicit requirement for Equ.(15) is that b_j is more ambiguous than b_i. After computing the messages passed from its neighbors, the belief of node b_j at the t-th iteration is updated by

$$\Psi^t_j(l_j) \;\propto\; P(b_j|l_j) \prod_{b_k\in\mathbb{N}(b_j)} \Phi^t_{k\to j}(l_j). \tag{16}$$

The belief for l_i can be viewed as an approximation of its marginal distribution $P(l_i|B)$.

C. Iterative Belief Reconfiguration

The BP algorithm is sometimes limited by stuck in unsatisfied local minima. One possible improvement is to reject the invalid solutions by imposing some constraints during inference. For example, Kschischang et al. [7] adopted high-order factors into energy potentials. However, this will greatly complicate the computation of messages, and is unsuitable for the tracking task.

In this work, we propose to utilize global constraints to verify each converged solution and re-activate the inference by reconfiguring the beliefs of nodes. Global constraints reflect intuitive and effective characteristics of object trajectories. For example, tracking targets with very small or very large size probably results from the trajectory being over-segmented or under-segmented, respectively. Specifically,

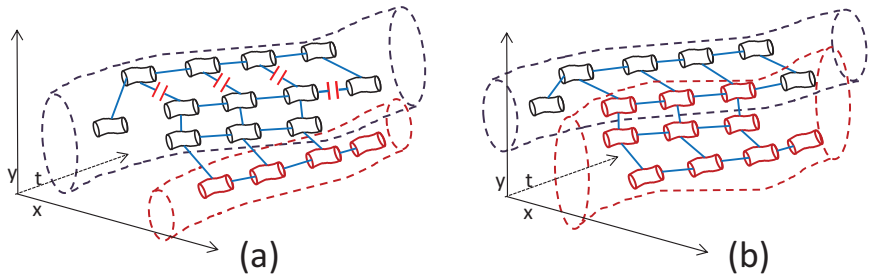

Fig. 3. An illustration of our Merge-and-Spilt operation to drive the solution reconfiguration. The red cuts of edges indicate the edges been turned off in the operation.

we define the global constraints on an object trajectory Γ_n as bivariate Gaussian distributions on averaged object size in time intervals

$$P(\Gamma_n) = \prod_z P(\Lambda_{n,z}) = \prod_z \mathbf{G}([\bar{w}_{n,z}\ \bar{h}_{n,z}]\,|\,\mu, \sigma^2), \tag{17}$$

where z denotes the z-th time interval of the trajectory, $\Lambda_{n,z}$ the region of n-th tracking target in time interval z, $[\bar{w}_{n,z}\ \bar{h}_{n,z}]$ the size of the bounding box of $\Lambda_{n,z}$ averaged over the time interval. The parameters μ and σ are tuned to fit the size of most objects in the dataset. In the experiments, we set the time interval as 8 frames and determine an object trajectory violates global constraints if there exists one time interval where the probability is less than 0.01.

For a converged solution, we first identify the most problematic partitioned trajectory, i.e. Γ_n violating global constraints and with smallest $P(\Gamma_n)$. Then we design a corresponding operator, i.e. merge-and-split, to reconfigure the solution by correcting Γ_n over the graph.

Merge-and-Split. For the case that we identify the problematic trajectory Γ_n by its very small region, i.e. the size of $\Lambda_{n,z}$ is smaller than a threshold, we scatter every node $b_i \in \Gamma_n$ and merge them with their neighbors nodes according to the affinities specified by the edge connections. The belief vector of b_i is then revised by setting 0 to the n-th bin. Note that we need to re-normalize the beliefs of all vertices over the graph accordingly. In other case, when one region of the trajectory Γ_n is too large, we split the vertices of Γ_n into two subsets. The vertices in one subset remain unchanged while the rest vertices are merged to their neighbors. The belief of each changed vertex is also need to be revised, i.e. the n-th of the belief vector is set as 0. In the implementation, we make the split using Normalized-Cuts [25] on positive and negative edges. An illustration is shown in Fig. 3.

Once the solution is reconfigured into a new partition, the Priority-based Belief Propagation will be re-performed to calculating the beliefs over the graph.

These above steps iterate until the target energy converges finally, i.e. the solution will not be reconfigured. The overall procedure is summarized in Algorithm 1.

Algorithm 1: Sketch of the RBP algorithm

Input: Video bundles B

Output: Bundle labels L

Initialize: $\Phi_{i\to j}(l_i) = 1$, $\Psi_i(l_i)$ by mechanism proposed in Section V-A, $l_i = \max_{l_i} \Psi_i(l_i)$;

repeat

  Reconfigure beliefs and labels for bundles within the problematic object trajectory ;

  for $t = 1$ *to* T_2 **do**

    repeat

      Prioritize bundle b_i according to its level of ambiguity ;

      foreach *more ambiguous neighbors of* b_i **do**

        Compute message from bundle b_i to bundle b_j by Equ.(15) ;

        Update belief for bundle b_j by Equ.(16) ;

      end

    until *all bundles pass messages*;

  end

  Assign each bundle $b_i \in B$ the label l_i with the max belief ;

until L *satisfies global constraints or algorithm iterates over* T_1 *times*;

VI. Experiment

Datasets and settings. We evaluate our algorithm for tracking multiple interacting and deforming agents on *figment* (figure untanglement) dataset, where a basketball court is filmed from a freely moving camera. This dataset, consisting of 18 challenging clips of 1080 frames total and 50-80 frames each, is a standard benchmark for detection-free tracking [15], and also used for video annotation testing [5].

All the parameters are fixed in the experiments. Point tracks are embedded into a $200 \sim 250$ dimensional manifold and further grouped into video bundles. The number of superpixel clusters K is between $5 \sim 12$. We set the maximum reconfiguration times $T_1 = 10$ and BP iterations a maxima of $T_2 = 15$. For the global constraints, the mean μ is fixed as $\mu = [45\ 90]$ and the standard variation a diagonal $\sigma_{i,i} = 10$ matrix.

We first compare the propose method with three detection-free tracking frameworks [26], [15], [16]. All these works use clustering of point tracks. The metrics used to measure the tracking quality are shown in Table I. For fair comparisons, the calculation of metrics follows the standard process introduced

TABLE I

EVALUATION METRICS FOR COMPARISONS WITH DETECTION-FREE TRACKING FRAMEWORKS.

Metric	Definition
Clustering error	The average percentage of pixels with false labels.
Per-region Clustering error	The average percentage of pixels with false labels per person mask.
Over-segmentation	The average number of object trajectories assigned to each mask.
Leakage	The percentage of trajectories exist leaking (overlap with other masks greater than 50% of theirs assigned masks) at one frame.
Recall	The percentage of recalled pixels for each labeled mask by the single best trajectory.
Tracking time	The percentage of frames whose recall for each mask is above 20% and averages across all masks.

TABLE II

QUANTITATIVE RESULTS AND COMPARISONS WITH DETECTION-FREE TRACKING METHODS. *Our-full*: OUR METHOD (VIDEO BUNDLE REPRESENTATION AND RBP ALGORITHM). *Our-1,2,3*: THREE SIMPLIFIED VERSION OF OUR APPROACH.

	Clustering error	Per-region clustering error	Over-segmentation	Leakage	Recall	Tracking time
Our-full	**4.24%**	**18.18%**	**1.01**	**15.84%**	**55.23%**	76.45%
Our-1	6.27%	23.60%	0.92	19.27%	46.50%	64.99%
Our-2	6.78%	37.79%	1.12	23.89%	43.31%	60.41%
Our-3	9.04%	33.15%	1.62	37.44%	33.27%	41.80%
[16]	7.90%	18.47%	1.5	19.55%	33.28%	82.29%
[15]	4.73%	20.32%	1.57	16.52%	31.07%	75.13%
[26]	20.74%	86.43%	0	81.55%	0.46%	1.03%

in [15]: (1) All metrics are computed by discarding top and bottom 10% of values and averaging over the remaining ones. (2) PRCE for a mask is set to 100% if it is missed to be assigned a label. (3) Recall for a leaking cluster is set to 0. (4) Recall and tracking time are computed by dilating each trajectory with a radius of 8 pixels. As Table II reports, our framework outperforms all the competing methods on all metrics except TT. Fig. 5 exhibits some tracking results, and we attach more video results in the supplemental materials.

We further compare our method with two detection-based tracking method [9], [17], which obtain object hypothesis by running human detectors [21] and [18], respectively. We adopt their publicly available codes where we tuned the parameters to achieve the best performance on *figment*. [9] poses tracking as a network flow problem that is solved approximately via dynamic programming. Note [17] also utilize point tracks as low-level cues. In each frame, the bounding boxes for each target are simply localized as

the cluster center of all pixels within the corresponding label. We calculate an one-to-one assignment of object hypotheses to groundtruth and utilize the widely used CLEAR MOT metrics [14] for evaluation, as shown in Table III. Note the Multiple Object Tracking Accuracy*(MOTA)* combines three error ratios into a single number. The quantitative results are reported in Table IV and some representative results are shown in Fig. 5. From the results, we observe that our method recovers substantially more trajectories with the higher tracking accuracy (reaching 56.71% MOTA). The competing methods do not work well due to the unreliable detections.

TABLE III

EVALUATION METRICS FOR COMPARISONS WITH DETECTION-BASED TRACKING FRAMEWORKS.

Metric	**Definition**
Miss detection	The average percentage of groundtruth objects failed to be detected.
False positive	The average percentage of detected objects does not belong to ground-truth objects.
ID-switch	The average percentage of times a groundtruth object changes its assigned identity.
MOTA	Accuracy $= 1 -$ MD $-$ FP $-$ ID-sw.

Component evaluation. We further present some empirical analysis to show the benefits of some components in our framework. Three additional results are generated by disabling some components of our framework, as Table II reports. *Our-1* generates trajectories on video bundles using priority-based belief propagation [4] (i.e. to replace the RBP algorithm). *Our-2* generates trajectories by performing spectral clustering on video bundles, (i.e. without the spatio-temporal inference). *Our-3* performs spectral clustering on point tracks directly to generate trajectories, (i.e. without extracting video bundles). In addition, we analyze the energy convergence for the RBP algorithm under a certain scenario. Note that the energy can be derived by the posterior probability. In Fig. 4 (right), we visualize the energy decreasing during inference and compare with the traditional BP algorithm. It is clearly shown that our algorithm can achieve better convergence while the BP algorithm stops after 4 iterations. As the line of energy implies,

TABLE IV

QUANTITATIVE RESULTS AND COMPARISONS WITH DETECTION-BASED TRACKING METHODS.

	Miss detection	False positive	ID-switch	MOTA
Our Method	26.09%	15.13%	2.07%	**56.71%**
[17]	50.95%	16.41%	0.43%	32.64%
[9]	89.19%	0.18%	4.46%	6.17%

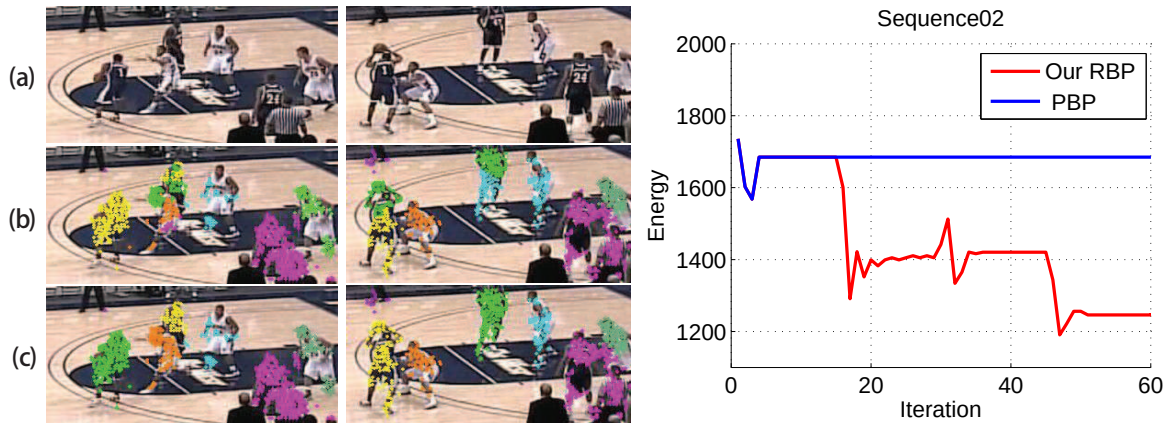

Fig. 4. Tracking results and analysis for comparing different inference algorithm. Left figure: (a) shows source frames. The tracking results by the traditional BP and the proposed RBP are exhibited in (b) and (c), respectively. The colors represent the tracking labels on pixels. Right figure: the red line represent the energy of the RPB algorithm, and the blue for the traditional Priority-based BP.

RBP avoids to get stuck on unsatisfied local minima, and re-activate inference by the reconfiguration operations. A few tracking results are highlighted in Fig. 4 (left).

Efficiency. Our implementation is coded in Matlab on an Intel I5 3.0 GHz PC with 4GB memory. On average, the runtime speed for bundle generation is $250ms$ per sequence, given the extracted point tracks. The inference averagely costs $5min \sim 8min$ per video sequence, related to the complexity of video content.

VII. Conclusion

This paper studied a novel video tracking framework in the context of object detectors been limited. Our approach constructed a spatial-temporal graph consisting of the video bundles by exploiting multiple cues of motion and appearance, and generated the trajectories by a novel robust belief propagation algorithm. The experiments and comparisons to the state-of-arts demonstrated the effectiveness of our framework on very challenging scenarios.

References

[1] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. pages 466–479, 2010.

[2] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. pages 1273–1280, 2011.

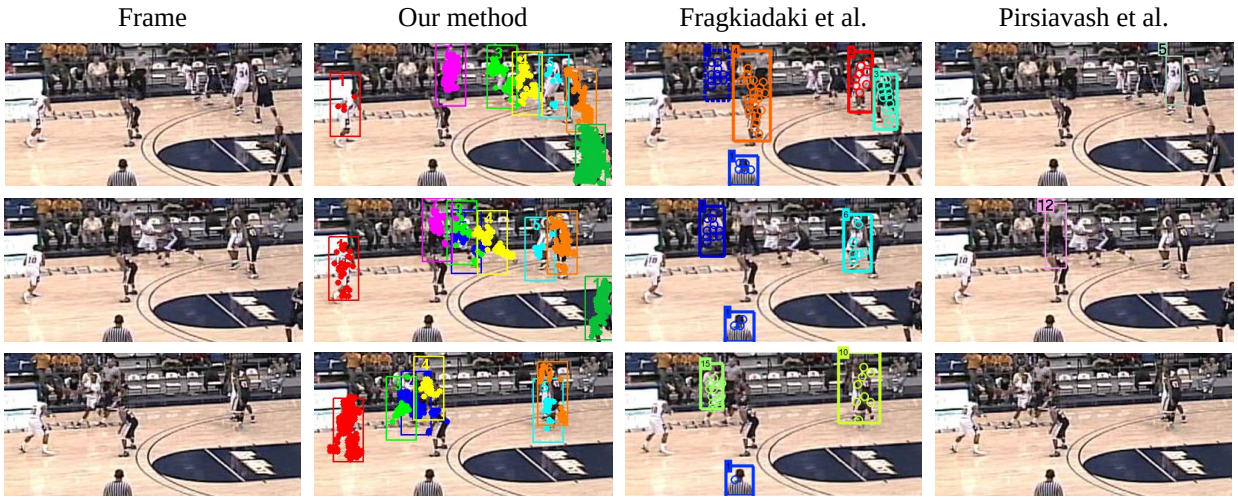

Fig. 5. Segmentation and tracking results on the figment dataset.

[3] A. Basharat, Y. Zhai, M. Shah. Content based video matching using spatiotemporal volumes. *CVIU*, 110(3):360–377, 2008.

[4] A.K. Kc and C. De Vleeschouwer. Prioritizing the propagation of identity beliefs for multi-object tracking. *In Proc. BMVC*, 2012.

[5] C. Vondrick, D. Ramanan, and D. Patterson. Efficiently scaling up video annotation with crowdsourced marketplaces. *In Proc. ECCV*, 2010.

[6] E.J. Keogh and M.J. Pazzani. Scaling up dynamic time warping for datamining applications. *In Proc. KDD*, 2000.

[7] F.R. Kschischang, B.J. Frey and H.A. Loeliger. Factor graphs and the sum-product algorithm. *TIT*, 47(2):498–519, 2001.

[8] H. Pirsiavash, D. Ramanan and C.C. Fowlkes. A linear programming approach for multiple object tracking. *In Proc. CVPR*, 2007.

[9] H. Pirsiavash, D. Ramanan and C.C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. *In Proc. CVPR*, 2011.

[10] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 33(9):1806–1819, 2011.

[11] J. Lezama, K. Alahari, J. Sivic and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. *In Proc. CVPR*, 2011.

[12] J. Shi and C. Tomasi. Good features to track. *In Proc. ICCV*, 1994.

[13] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 75(2):888–905, 1997.

[14] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *J. Image Video Process*, 2008.

[15] K. Fragkiadaki and J. Shi. Detection-free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. *In Proc. CVPR*, 2011.

[16] K. Fragkiadaki, G. Zhang and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. *In Proc. CVPR*, 2012.

[17] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi. Two granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. *In Proc. ECCV*, 2012.

[18] L. Bourdev, S. Maji, T. Brox and J. Malik. Detecting people using mutually consistent poselet activations. *In Proc. ECCV*, 2010.

[19] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. *In Proc. CVPR*, 2008.

[20] M. Andriluka, S. Roth and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. *In Proc. CVPR*, 2008.

[21] P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32(9):1627–1645, 2010.

[22] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

[23] S. Pellegrini, A. Ess, and L.V. Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. *In Proc. ECCV*, 2010.

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

[25] S.X. Yu and J. Shi. Understanding popout through repulsion. *In Proc. CVPR*, 2001.

[26] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. *In Proc. ECCV*, 2010.

[27] V. Badrinarayanan, P. Pérez, F. Le Clerc and Lionel Oisel. Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues. *In Proc. ICCV*, 2007.

[28] X. Liu, L. Lin, S.C. Zhu and H. Jin. Trajectory parsing by cluster sampling in spatio-temporal graph. *In Proc. CVPR*, 2009.

[29] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. *In Proc. CVPR*, 2007.

[30] Q. Yu and G. G. Medioni. Multiple-target tracking by spatiotemporal monte carlo markov chain data association. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2196–2210, 2009.