Multi-view People Tracking via Hierarchical Trajectory Composition

Yuanlu Xu[1], Xiaobai Liu[2]*, Yang Liu[1] and Song-Chun Zhu[1]
[1]Dept. Computer Science and Statistics, University of California, Los Angeles (UCLA)
[2]Dept. Computer Science, San Diego State University (SDSU)
yuanluxu@cs.ucla.edu, xiaobai.liu@mail.sdsu.edu, yangliu2014@ucla.edu, sczhu@stat.ucla.edu

Abstract

This paper presents a hierarchical composition approach for multi-view object tracking. The key idea is to adaptively exploit multiple cues in both 2D and 3D, e.g., ground occupancy consistency, appearance similarity, motion coherence etc., which are mutually complementary while tracking the humans of interests over time. While feature online selection has been extensively studied in the past literature, it remains unclear how to effectively schedule these cues for the tracking purpose especially when encountering various challenges, e.g. occlusions, conjunctions, and appearance variations. To do so, we propose a hierarchical composition model and re-formulate multi-view multi-object tracking as a problem of compositional structure optimization. We setup a set of composition criteria, each of which corresponds to one particular cue. The hierarchical composition process is pursued by exploiting different criteria, which impose constraints between a graph node and its offsprings in the hierarchy. We learn the composition criteria using MLE on annotated data and efficiently construct the hierarchical graph by an iterative greedy pursuit algorithm. In the experiments, we demonstrate superior performance of our approach on three public datasets, one of which is newly created by us to test various challenges in multi-view multi-object tracking.

1. Introduction

Multi-view multi-object tracking has attracted lots of attentions in the literature [22]. Tracking objects from multiple views is by nature a composition optimization problem. For example, a 3D trajectory of a human can be hierarchically decomposed into trajectories of individual views, trajectory fragments, and bounding boxes. While existing trackers have exploited the above principles more or less, they enforced strong assumptions over the validity of a particular cue, e.g. appearance similarity [1], motion consistency [9], sparsity [30, 50], 3D localization coincidence [24], etc., which are not always correct. Actually, different cues may dominate different periods over object trajectories, especially for complicated scenes. In this paper, we are interested in automatically discovering the optimal compositional hierarchy for object trajectories from various cues, in order to handle a wider variety of tracking scenarios.

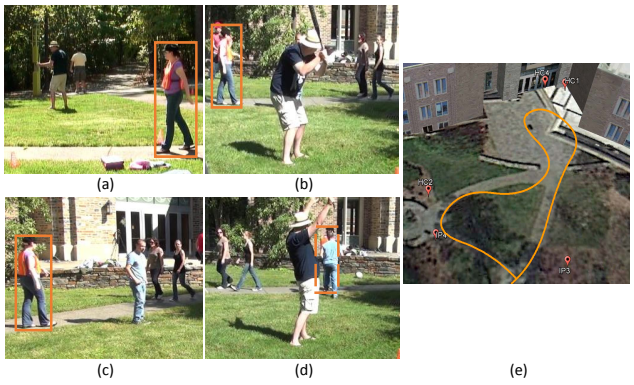

Figure 1. An illustration of utilizing different cues at different periods for the task multi-view multi-object tracking.

As illustrated in Fig. 1, suppose we would like to track the highlighted subject and obtain its complete trajectory (e). The optimal strategy for tracking may vary over space and time. For example, in (a), since the subject shares the same appearance within certain time period, we apply an appearance based tracker to get a 2D tracklet; in (b) and (c), since the subject can be fully observed from two different views, we can group these two boxes into a 3D tracklet by testing the proximity of their 3D locations; in (d), since the subject is fully occluded in this view, we consider sampling its position from the 3D trajectory curve constrained by background occupancy.

In this work, we formulate multi-view multi-object tracking as a structure optimization problem described by a hierarchical composition model. As illustrated in Fig. 2, our objective is to discover composition gradients of each

*Correspondence author is Xiaobai Liu. This work is supported by DARPA SIMPLEX Award N66001-15-C-4035, ONR MURI project N00014-16-1-2007, and NSF IIS 1423305.

object in the hierarchical graph. We start from structureless tracklets, i.e., object bounding boxes, and gradually compose them into tracklets of larger size and eventually into trajectories. Each trajectory entity may be observed in single view or multiple views. The composition process is guided by a set of criteria, which describe the composition feasibility in the hierarchical structure.

Each criterion focuses on one certain cue and in fact is equivalent to a simple tracker, e.g., appearance tracker [29, 45], geometry tracker [35], motion tracker [2], etc., which groups tracklets of the same view or different views into tracklets of larger sizes. Composition criteria lie in the heart of our method: feasible compositions can be conducted recursively and thus the criteria can be efficiently utilized.

To infer the compositional structure, we divest MCMC sampling-based algorithms due to their heavy computation complexity. We approximate the hierarchy by a progressive composing process. The composition scheduling problem is solved by an iterative greedy pursuit algorithm. At each step, we first greedily find and apply the composition with maximum probability and then re-estimate parameters for the incremental part.

In the experiments, we evaluate the proposed method on a set of challenging sequences and the results demonstrate superior performance over other state-of-the-art approaches. Furthermore, we design a series of comparison experiments to systematically analyze the effectiveness of each criterion.

The main contributions of this work are two-fold. Firstly, we re-frame multi-view multi-object tracking as a hierarchical structure optimization problem and present three tracklet-based composition criteria to jointly exploit different kinds of cues. Secondly, we establish a new dataset to cover more challenges, to present richer visual information and to provide more detailed annotations than existing ones.

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

2. Related Work

Our work is closely related to the following four research streams.

Multi-object tracking has been extensively studied in the last decades. In the literature, the tracking-by-detection pipeline [47, 20, 33, 41, 7, 8] attracts wide-spreaded attentions and acquires impressive results, thanks to the considerable progress in object detection [12, 37, 34], as well as in data association [48, 32, 6]. In particular, network flow based methods [32, 6] organize detected bounding boxes into directed multiple Markov chains with chronological order and pursue the trajectory as finding pathes. Andriyenko et al. [2] propose to track objects in discrete space and use splines to model trajectories in continuous space. Our approach also follows this pipeline but considers bounding boxes as structureless elements. With preliminary associations to preserve locality, we can better explore the nonlocal properties [23] of trajectories in the time domain. For example, tracklets with evident appearance similarities can be grouped together without considering the time interval.

Multi-view object tracking is usually addressed as a data association problem across cameras. The typical solutions include, homography constraints [24, 4], ground probabilistic occupancy [14], network flow optimization [42, 6, 25], marked point process [38], joint reconstruction and tracking [19], multi-commodity network [36] and mutiview SVM [49]. All these methods have certain strong assumptions and thus are restricted to certain specific scenarios. In contrast, we are interested in discovering the optimal composition structure to obtain complete trajectories in a wide variety of scenarios.

Hierarchical model receives heated endorsement for its effectiveness in modeling diverse tasks. In [17], a stochastic grammar model was proposed and applied to solve the image parsing problem. After that, Zhao et al. [51] and Liu et al. [27] introduced generative grammar models for scene parsing. Pero et al. [31] further built a generative scene grammar to model the constitutionality of Manhattan structures in indoor scenes. Ross et al. presented a discriminative grammar for the problem of object detection [15]. Grosse et al. [16] formulated matrix decomposition as a structure discovery problem and solved it by a context-free grammar model. In this paper, our representation can be analogized as a special hierarchical attributed grammar model, with similar hierarchical structures, composition criteria as production rules, and soft constraints as probabilistic grammars. The difference lies in that our model is fully recursive and without semantics in middle levels.

Combinatorial optimization receives considerable attentions in the surveillance literature [43]. When the solution space is discrete and the structure cannot be topologically sorted (e.g., loopy graphs), there comes the problem of combinatorial optimization. Among all the solutions, MCMC techiques are widely acknowledged. For example, Khan et al. [24] integrated the MCMC sampling within the particle filer tracking framework. Yu et al. [46] utilized the single site sampler for associating foreground blobs to trajectories. Liu et al. [28] introduced a spatial-temporal graph to jointly solve the region labeling and object tracking problem by Swendsen-Wang Cut [5]. In this work, though facing a similar combinatorial optimization problem, we propose a very efficient inference algorithm with acceptable trade-off.

3. Representation

In this section, we first introduce the compositional hierarchy representation, and then discuss the proposed problem formulation for multi-view multi-object tracking.

3.1. Hierarchical Composition Model

Given an input sequence containing videos shot by multiple cameras, we follow a default tracking-by-detection pipeline and apply [34] to obtain detected bounding boxes. After that, we associate them into short trajectory fragments, i.e., tracklets, similar to [20, 40]. Tracklets preserve better local properties of appearance and motion as well as better robustness against errors and noises, compared with bounding boxes.

We denote a tracklet as O, which contains the appearance and geometry information over a certain period of time:

$$O = \{(a_i, l_i, t_i) : \ i = 1, 2, \ldots, |O|\}, \tag{1}$$

where a_i is the appearance feature, l_i the location information (i.e., 2D bounding box and 3D ground position) and t_i the time stamp. Note that the 3D ground position is calculated by projecting the foot point of the 2D bounding box onto the world reference frame. For convenience, we denote the start time and end time of a tracklet by t^s and t^e, respectively. We further augment a set of states $x(O)$ for each tracklet O

$$x(O) = \{\omega_i : \ i = 1, \ldots, |O|\}, \tag{2}$$

where $\omega_i \in \{1, 0\}$ indicates the state of visibility/invisibility on the 3D ground plane at time t_i. $x(0)$ describes the sparsity of a trajectory and can be utilized to enforce the consistency of object appearing and disappearing over time.

As shown in Fig. 2, we organize the scene as a compositional hierarchy $\mathbb{G}$ to recover the trajectory for each object in both single views and 3D ground. The compositional hierarchy $\mathbb{G}$ is denoted as

$$\mathbb{G} = (V_N, V_T, S, X), \tag{3}$$

where V_T denotes the set of terminal nodes, V_N indicates the set of non-terminal nodes, S is the root node representing the scene, and X represents the set of states of both terminal and non-terminal nodes.

A non-terminal node O is constructed by composing two nodes O_1 and O_2 together, that is

$$O \leftarrow f(O_1, O_2), \quad g_i(x(O)) = f_i(x(O_1), x(O_2)), \tag{4}$$

where $g_i(\cdot)$ and $f_i(\cdot)$ are associated operations on states. Note that $g_i(\cdot)$ and $f_i(\cdot)$ can assign states in either bottom-up or top-down direction, which act like functions of passing messages.

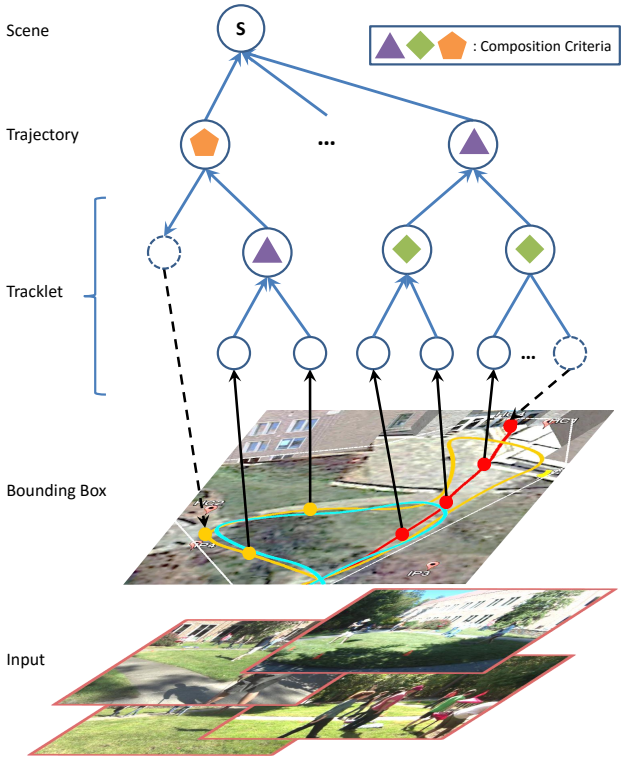

Figure 2. An illustration of the hierarchical compositional structure.

3.2. Bayesian Formulation

According to Bayes' rule, we can solve the problem of inferring the hierarchical composition model by maximizing a posterior, that is,

$$\mathbb{G}^* = \arg\max_{\mathbb{G}} p(\mathbb{G}|I) \propto \arg\max_{\mathbb{G}} p(I|\mathbb{G}) \cdot p(\mathbb{G}), \tag{5}$$

where I denotes the input video data.

Prior. Due to the property of hierarchy, we can further factorize the prior $p(\mathbb{G})$ as

$$p(\mathbb{G}) = \prod_{O_i \in V_N} p(x(O_i)) \prod_k p_k^{cp}(O_{i1}, O_{i2})^{\delta_i == k}, \tag{6}$$

where δ_i is an indicator for the type of criterion used in composition, and O_{i1} and O_{i2} are two children nodes of tracklet O_i.

$p(x(O))$ is a unary probability defined on the state of O. We employ a simple Ising/Potts model to penalize the discontinuity of the trajectory, i.e.,

$$p(x(O)) \propto \exp\{-\beta \sum_{i=1}^{|O|-1} \mathbf{1}(\omega_i \neq \omega_{i+1})\}, \tag{7}$$

where β is a coefficient. $p(x(O))$ in fact constrain the number of times a trajectory switches between visible and invisible.

$p_k^{cp}(O_i, O_j)$ represents the composition probability using the k-th type of cue. We will discuss details about of composition criteria in Section 3.3.

Likelihood. The video data I is only dependent on the terminal nodes V_T and can be further decomposed as

$$p(I|\mathbb{G}) = \left(\prod_{O_i \in V_T} \prod_{a_j \in O_i} p^{fg}(a_j) \right) \cdot \prod_{a_j \in I \setminus V_T} p^{bg}(a_j) \\ = \prod_{O_i \in V_T} \prod_{a_j \in O_i} \frac{p^{fg}(a_j)}{p^{bg}(a_j)} \cdot \prod_{a_j \in I} p^{bg}(a_j), \tag{8}$$

where $p^{fg}(\cdot)$ and $p^{bg}(\cdot)$ are foreground and background probabilities, respectively. The second term $\prod_{a_j} p^{bg}(a_j)$ measures the background probability over the entire video data and thus can be treated as a constant, and the first term measures the divergence between foreground and background, which can be analogous to a probabilistic foreground/background classifier. We use the detection scores to approximate this log-likelihood ratio.

3.3. Composition Criteria

In this section, we introduce details of the proposed composition criteria.

Appearance Coherence. Instead of using traditional descriptors (e.g., SIFT, color histograms, MSCR) to measure the appearance discrepancy, we employ the powerful DCNN to model people's appearance variations. Notice that most DCNNs are trained over generic object categories and insufficient to provide fine-grained level of information about peoples identities [44]. We therefore fine-tune the CaffeNet [21] using people image samples with identity labels. The new DCNN consists of 5 convolutional layers, 2 max-pooling layers, 3 fully-connected layers and a final 1000-dimensional output. The last two layers are discarded and replaced by random initializations. The output is new 1000 labels on people's identities. Note the training samples are augmented from unlabeled data and identity labels are obtained in an unsupervised way.

Similar to bag-of-words (BoW), our DCNN plays the role of a codebook, which codes a person image with common people appearance templates. We use this 1000-dimensional output as our appearance descriptor. Given two tracklets O_i and O_j, the appearance coherence constraint $p_1^{cp}(O_i, O_j)$ is defined as

$$p_1^{cp}(O_i, O_j) \propto \exp\{-\frac{\sum_{a_n \in O_i} \sum_{a_m \in O_j} \|a_n - a_m\|_2}{|O_i| \cdot |O_j|}\}. \tag{9}$$

$p_1^{cp}(O_i, O_j)$ actually measures the mean complete-link appearance dissimilarities among object bounding boxes belonging to two tracklets.

Geometry Proximity. Given tracklets from a single view or cross views, we first project them on the world reference frame to measure their geometric distances uniformly. However, considering tracklets with different time stamps and lengths, it is not a trivial task to determine whether the two given tracklets belong to the same object or not. The reason lies in: i) the time stamps of tracklet pairs might not be well aligned; ii) the localizations across views usually lead to remarkable amount of errors.

In order to address these issues, we introduce a kernel to measure these time series samples. The kernel $K(O_i, O_j)$ to measure the distance between two tracklets O_i and O_j is defined as the product of two kernel distances in space and time

$$K(O_i, O_j) = \sum_{(l_n, t_n) \in O_i} \sum_{(l_m, t_m) \in O_j} \frac{\phi_l(l_n, l_m) \cdot \phi_t(t_n, t_m)}{|O_i| \cdot |O_j|}, \tag{10}$$

where $\phi_l(l_n, l_m)$ and $\phi_t(t_n, t_m)$ are two RBF kernels between two points. We use different σ_l and σ_t values for the two kernels, respectively. This new kernel acts like a sequential convolution filter and takes both spatial and temporal proximities into consideration.

Given a set of training samples D,

$$D = \{(O_i, O_j, y_n) : \ n = 1, \ldots, |D|\}, \tag{11}$$

where $y_n \in \{1, 0\}$ indicates whether or not the two tracklets O_i and O_j belong to the same identity, we can train a kernel SVM with the energy function

$$\min_w \frac{1}{2} <w, w> + C \sum_n \max(0, 1 - y_n <w, K(O_i, O)>), \tag{12}$$

where C is a regularization factor.

We therefore interpret the normalized classification margin as the composition probability $p_2^{cp}(O_i, O_j)$.

Motion Consistency. We model the motion information of a tracklet O as a continuous function of its 3D ground positions l w.r.t. time t, i.e., $l = \tau(t)$. We define a constraint on two tracklets that they can be interpreted with the same motion function. However, finding this motion pattern is a challenging problem. The reason lies in two-fold: i) inaccurate 3D positions due to perspective effects, detection errors and false alarms; ii) missing detections and object inter-occlusions in certain views, especially for crowded scenarios. In this paper, we address these issues in the following two aspects.

Firstly, we employ the b-spline function to represent the motion pattern of the trajectory. B-spline functions can enforce high-order smoothness constraints, which enables learning from sparse and noisy data. Considering a tracklet O with 3D positions $\{l_i : i = 1, \ldots, |O|\}$, starting time t^s and ending time and t^e, the spline function $\tau(t)$ uses some quadratic basis functions $B_k(t)$, and represents the motion

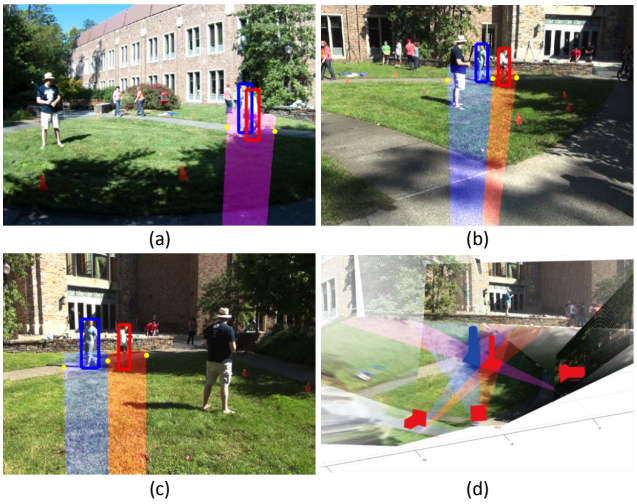

Figure 3. An illustration of finding feasible regions (polygons) for interacting people.

path as a linear combination of $B_k(t)$:

$$\begin{aligned} \tau(t) &= \sum_k \alpha_k B_k(t), \\ s.t.\ \ \tau''(t^s) &= \tau''(t^e) = 0, \end{aligned} \tag{13}$$

where $\tau''(t)$ denotes the second derivative of $\tau(t)$. The constraints enforce zero curvature at the starting and the ending point.

Secondly, we take advantages of the multi-view setting and derive feasible regions for object 3D positions to further confine the fitted motion curve. As illustrated in Fig. 3, given bounding boxes of a single object in the views (a), (b) and (c), we first perform exhaustive search to find the two anchor points (yellow dots in the image) along two sides of the foot position of each object. An anchor point is defined as a position where the surrounding 8×8 area contains most of background regions. Note that we generate background masks by GMM background modeling.

Once obtaining all the anchor points for an object, we can find the union area Ω, i.e., a polygon on the world ground plane, as shown the shaded area in (d). These polygons serve as additional localization feasibility constraints on the motion pattern. That is, the spline fitting is formulated as minimizing the following objective function:

$$\begin{aligned} \min_{\alpha_k, B_k}\ E(O_i, O_j) = & \sum_{(l_n, t_n) \in O_i \cup O_j} \left(l_n - \sum_k \alpha_k B_k(t_n) \right)^2, \\ & s.t. \quad \alpha_k B_k(t_n) \in \Omega_n. \end{aligned} \tag{14}$$

This is a constrained convex programming problem considering that all polygons are convex. We refer the readers to find more details about b-spline and robust fitting algorithms in [10].

The probability $p_3(O_i, O_j)$ is defined upon the averaged residuals for spline fitting, i.e.,

$$p_3^{cp}(O_i, O_j) \propto \exp\{-\frac{E(O_i, O_j)}{|O_i \cup O_j|}\}. \tag{15}$$

4. Learning and Inference

In this section, we first discuss the learning procedure for our constraints and then introduce how to infer the hierarchical compositional structure.

4.1. Learning Constraints

Appearance Coherence. Even for fine-tuning a DCNN, fair amount of training samples are required. We therefore augment the training data by external samples from public people detection datasets, e.g., CaltechPedestrians, NICTA, ETH and TUD-Brussels. The augmented training set contains around 30,0000 samples of cropped people images. We resize all the samples to 128×256 and horizontally flip them to double the training set size. And then we extract dense HSV color histograms with 16 bins from 16×16 non-overlapping patches for each image. The computed histograms are concatenated into a 6144-dimensional feature vector. We perform K-means clustering on the data and obtain 1000 clusters. Each cluster is regarded as a class and we utilize them to fine tune our DCNN. In general, the fine-tuning process converges after 100000 iterations and costs about 8 hours.

Geometry Proximity. Given the training data and groundtruth of a scenario, we first generate initial tracklets and then associate them with the groundtruth. A tracklet is treated as a fragment of a groundtruth trajectory if more than 50% of its bounding boxes are correctly assigned (i.e., hit/miss cutoff with 50% IoU ratio). The training data set D can thus be constructed using tracklets from the same trajectory as positive pairs samples and those from different trajectories as negative pairs. We learn the parameters of our kernel SVM for each pair of views (including self-to-self). The kernel parameters σ_l and σ_t are also tuned by cross-validation.

Note we also estimate the normalization constant for each constraint $p_k^{cp}(O_i, O_j)$ using the training data.

4.2. Inferring Hierarchy

Our objective is to find a compositional hierarchy $\mathbb{G}$ by maximizing the posterior probability formulated in Equation (5). The optimization algorithm should accomplish two goals: i) composing hierarchical structures, and ii) estimating states for terminal and non-terminal nodes.

The main challenge in optimizing Equation (5) lies in the size of the solution space. For example, if there are n terminal nodes, even a single group can be formed in 2^{n-1} different ways, which is exponential. Although MCMC sampling-based algorithms [28, 43] are favored to

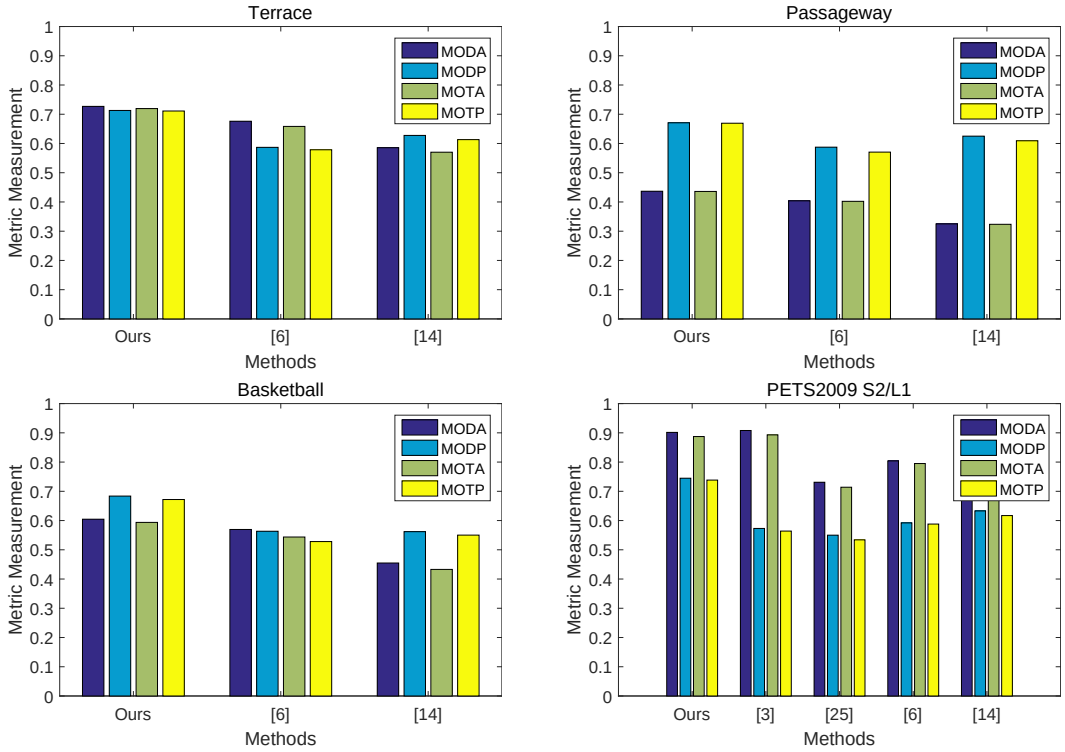

Figure 4. Comparison charts using CLEAR metrics on EPFL and PETS 2009 datasets.

solve such kinds of combinatorial optimization problems, they are typically computationally expensive and difficult to converge, especially for our case, with thousands of terminal nodes and numerous possible compositions.

Hereby, we approximate the construction of the hierarchical structure by a progressive composing process. In the beginning, given a set of initial tracklets V_T, we initialize the state $\omega_i \in x(O)$ for each tracklet O as visible. We then enumerate all the tracklets over all composition criteria, and find two tracklets O_i and O_j with maximum probability to be composed into a new tracklet O_n, that is,

$$\max_{O_i,O_j,\delta_n} p(x(O_n)) \prod_k p_k^{cp}(O_i,O_j)^{\delta_n==k}, \qquad (16)$$

where δ_n is an indicator for which cue is selected. We then group these two tracklets O_i and O_j together, and create their parent node O_n.

The states for this newly merged node O_n are re-estimated by

$$\begin{aligned} &x(O_n) = x(O_i) \cup x(O_j), \\ &t_n^s = \min(t_i^s, t_j^s), \;\; t_n^e = \max(t_i^e, t_j^e), \\ &|x(O_n)| = t_n^e - t_n^s + 1. \end{aligned} \qquad (17)$$

Note we set all the states of missing time stamps within the time scope $[t_n^s, t_n^e]$ to 0, i.e., invisible. This encourages future filling-in operations.

If a composition performed based on motion consistency constraint, we then fill in the missing fragments by interpolations, and create a corresponding tracklet $O_m \in V_T$. The new tracklet O_m will be naturally incorporated into the hierarchical structure by subsequent compositions.

We continue this process iteratively. If the maximum composition probability reaches the lower limit, we terminate the algorithm and connect all the top non-terminal nodes to the root node S. Each sub-tree connected to the root node is essentially an object trajectory.

5. Experiment

In this section, we first introduce the datasets and the parameter settings, and then show our experimental results as well as component analysis of the proposed approach.

5.1. Datasets and Settings

We evaluate our approach on three public datasets:

(i) **EPFL dataset**[1] [14]. We adopt the Terrace sequence 1, Passageway sequence and Basketball sequence in our experiments. In general, each sequence consists of 4 different views and films 6-11 pedestrians walking or running around, lasting 3.5-6 minutes. Each view is shot at 25fps and in a relatively low resolution 360×288.

(ii) **PETS 2009 dataset**[2] [13]. This dataset is widely used in evaluating tracking tasks and sequence S2/L1 is spe-

[1] Available at cvlab.epfl.ch/data/pom/

[2] Available at www.cvg.reading.ac.uk/PETS2009/a.html

Sequence	Method	MODA(%)	MODP(%)	MOTA(%)	MOTP(%)	MT(%)	PT(%)	ML(%)	IDSW	FRAG
Garden1	Our-full	**49.30**	72.02	**49.03**	71.87	31.25	62.50	6.25	299	200
	Our-3	44.63	72.35	44.36	72.20	18.75	68.75	12.50	296	202
	Our-2	42.10	71.08	41.69	70.97	12.50	75.00	12.50	448	296
	Our-1	41.21	71.06	37.21	70.94	12.50	75.00	12.50	4352	4390
	[6]	30.47	62.13	28.10	62.01	6.25	68.75	25.00	2577	2553
	[14]	24.52	64.28	22.43	64.17	0.00	56.25	43.75	2269	2233
Garden2	Our-full	**27.81**	71.74	**25.79**	71.59	21.43	78.57	0.00	94	73
	Our-3	23.39	71.13	22.50	71.08	14.29	85.71	0.00	92	72
	Our-2	18.76	70.20	17.27	70.12	14.29	78.57	7.14	142	97
	Our-1	17.68	70.12	10.24	70.11	14.29	78.57	7.14	700	733
	[6]	24.35	61.79	21.87	61.64	14.29	85.71	0.00	268	249
	[14]	16.51	63.92	13.95	63.81	14.29	78.57	7.14	241	216
Auditorium	Our-full	**20.84**	69.26	**20.62**	69.21	33.33	55.56	11.11	31	28
	Our-3	18.83	68.99	18.62	68.95	22.22	61.11	16.67	30	28
	Our-2	18.02	68.32	17.29	68.25	16.67	66.67	16.67	104	94
	Our-1	17.78	68.33	14.11	68.28	16.67	66.67	16.67	523	536
	[6]	19.46	59.45	17.63	59.29	22.22	61.11	16.67	264	257
	[14]	17.90	61.19	16.15	61.02	16.67	66.67	16.67	249	235
ParkingLot	Our-full	**24.46**	66.41	**24.08**	66.21	6.67	66.67	26.67	459	203
	Our-3	19.23	66.50	18.84	66.38	0.00	53.33	46.67	477	191
	Our-2	12.85	65.70	12.23	65.61	0.00	46.67	53.33	754	285
	Our-1	10.86	65.77	8.74	65.72	0.00	46.67	53.33	2567	2600
	[6]	14.73	58.51	13.99	58.36	0.00	53.33	46.67	893	880
	[14]	11.68	60.10	11.00	59.98	0.00	46.67	53.33	828	812

Table 1. Quantitative results and comparisons on CAMPUS dataset. Our-1, Our-2, Our-3 are three benchmarks set up for component evaluation. See text for detailed explanations.

cially designed for multi-view-based tasks. With 3 surveillance cameras and 4 DV cameras, 10 pedestrians are recorded entering, passing through, staying and exiting the pictured area. The video is downsampled to 720×576 and the frame rate is set to 7fps.

(iii) **CAMPUS dataset**. To cover more complete challenges not presented in existing databases, we design this dataset based on the idea of dense foreground (around 15-25 objects, frequent conjunctions and occlusions), complex scenarios (objects conducting diverse activities, dynamic background, interactions between objects and background), various object scales (tracking targets sometimes either too tiny or huge to be accommodated in certain cameras). We incorporate 4 sequences into this dataset: Garden 1, Garden 2, Auditorium and Parking Lot. Each sequence is shot by 3-4 high-quality DV cameras mounted around 1.5m-2m above ground and each camera covers both overlapping regions and non-overlapping regions with other cameras. The videos are recorded with frame rate 30fps and duration about 3-4 minutes. The resolution is preserved in 1920×1080, for better precision and richer information.

For all three datasets, videos in each sequence are synchronized. We fully annotate the groundtruth trajectories for all the videos in all the sequences using [39]. Note that we assign an unique ID for each object, whether it appears once or several times in the scene. Since the ultimate task of multi-view multi-object tracking is to discover the complete 3D trajectory of any targeted individual under a camera network, we believe uniquely assigned ID should be the groundtruth to fully evaluate the trackers, which poses higher requirements than conventional tracking tasks [22]. In experiments, we use the beginning 10% video data for training and the rest for testing.

All the parameters are fixed in the experiments. For object detection, we use the PASCAL VOC fine-tuned ZF net, score threshold 0.3 and NMS threshold 0.3, which obtains proper trade-off between the efficiency and effectiveness. As for tracklet initialization, we construct a graph with edges only connected among successive frames and within limited scale changes. That is, sizes of two successive bounding boxes should not change more than 25% larger or smaller, in either height or width. We then run the successive shortest path algorithm [32] to generate tracklets. Empirically, this produces short but identity consistent tracklets. $\beta = 0.05$ in the unary probability $p(x(O))$. The motion consistency constraint is conducted on tracklets with time interval no longer than 2 seconds, with the B-spline of order at most 3 and breaks at most 4. In the hierarchical composition, the lower limit is set to 0.2, which obtains good results.

5.2. Experimental Results

We employ the widely used CLEAR metrics [22], Multiple Object Detection Accuracy (MODA), Detection Precision (MODP), Tracking Accuracy (MOTA) and Tracking Precision (MOTP) to measure three kinds of errors in tracking: false positives, false negatives and identity switches. Besides, we also report the percentage of *mostly tracked*

Figure 5. Results generated by the proposed method on CAMPUS, EPFL and PETS 2009 datasets.

(MT), *partly tracked* (PT) and *mostly lost* (ML) groundtruth (referring to [26]), as well as the number of identity switches (IDSW) and fragments (FRAG). Hit/miss for the assignment of tracking output to groundtruth is set to a threshold of Intersection-over-Union (IoU) ratio 50%.

We compare the proposed approach with 4 state-of-the-arts methods: Probabilistic Occupancy Map (POM) [14], K-Shortest Path (KSP) [6], Branch-and-Price [25] and Discrete-Continuous Optimization [3]. We adopt the public code of POM detection and implement the data association algorithms "DP with appearance" [14] and KSP [6] according to their descriptions. The reported metrics for comparing methods are quoted on PETS 2009 dataset from [11] and computed on the rest by conducting experiments.

Quantitative evaluations on EPFL and PETS 2009 datasets is shown in Fig. 4 and CAMPUS dataset in Table 1, as well as qualitative results in Fig. 5. From the results, our method demonstrates superior performance over the competing methods. We can also observe the proposed method acquires significant margins on MODP, MOTP, IDSW and FRAG, which indicates two empirical conclusions: i) detection-based tracklet initialization is more beneficial to object overall localization than foreground-blob-based methods which mainly concerns ground positions; ii) when it comes to occlusions, multiple cues (e.g., appearance, geometry, and motion) are all neccessary to keep the trajectory identity consistent, which has also been approved in [18]. Competing methods do not work well on CAMPUS dataset mainly due to their strong dependence on clear visibility of ground plane and uniform object size.

Component Analysis. We set up three benchmarks to further analyze the benefits of each production rule on CAMPUS dataset. *Our-1* outputs the initial tracklets directly, i.e., no composition performed; *Our-2* composes the hierarchy only using the appearance coherence criterion; *Our-3* further incorporates the geometry proximity criterion; *Our-full* employs all criteria proposed in this paper. From the results, it is apparent that each constraint contributes to a better hierarchical composition model.

Efficiency. Our method is implemented in MATLAB and runs on a desktop with Intel I7 3.0GHz CPU, 32GB memory and Nvidia GTX780Ti GPU. Given a 1080P sequence, the runtime on average is 15-20fps for object detection, 1000-1500fps for tracklet initialization, and 2-4fps for optimizing the hierarchical structure. Overall, the proposed algorithm obtains 1-3fps, which is related to the object density of the sequence. With proper code migration and optimization, e.g., batch processing, we believe the realtime processing can be achieved.

6. Conclusion

This paper studies a novel formulation for multi-view multi-object tracking. We represent object trajectories as a compositional hierarchy and construct it with probabilistic constraints, which characterize the geometry, appearance and motion properties of trajectories. By exploiting multiple cues and composing them with proper scheduling, our method handles challenges in multi-view multi-object tracking well. Furthermore, we will explore more powerful inter-tracklet relations and better composition algorithms in the future.

References

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

2006.

[2] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Proc. CVPR*, 2011.

[3] A. Andriyenko and K. Schindler. Discrete-continuous optimization for multi-target tracking. In *Proc. CVPR*, 2012.

[4] M. Ayazoglu, B. Li, C. Dicle, M. Sznaier, and O. Camps. Dynamic subspace-based coordinated multicamera tracking. In *Proc. ICCV*, 2011.

[5] A. Barbu and S. Zhu. Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *IEEE Trans. PAMI*, 27(8):1239–1253, 2007.

[6] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. multiple object tracking using k-shortest paths optimization. *IEEE Trans. PAMI*, 33(9):1806–1819, 2011.

[7] A. Dehghan, S. Assari, and M. Shah. Gmmcp-tracker:globally optimal generalized maximum multi clique problem for multiple object tracking. In *Proc. CVPR*, 2015.

[8] A. Dehghan, Y. Tian, P. Torr, and M. Shah. Target identity-aware network flow for online multiple target tracking. In *Proc. CVPR*, 2015.

[9] C. Dicle, O. Camps, and M. Sznaier. The way they move: tracking multiple targets with similar appearance. In *Proc. ICCV*, 2013.

[10] P. Eilers and B. Marx. Flexible smoothing with b-splines and penalties. *Statistical Science*, 11(2):89–121, 1996.

[11] A. Ellis, A. Shahrokni, and J. Ferryman. Pets2009 and winter-pets 2009 results: A combined evaluation. In *Proc. Winter-PETS Workshop*, 2009.

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

[13] J. Ferryman and A. Shahrokni. Pets2009: Dataset and challenge. In *Proc. Winter-PETS Workshop*, 2009.

[14] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE Trans. PAMI*, 30(2):267–282, 2008.

[15] R. B. Girshick, P. F. Felzenszwalb, and D. A. McAllester. Object detection with grammar models. In *Proc. NIPS*, 2011.

[16] R. B. Grosse, C. Sci, R. Salakhutdinov, W. T. Freeman, C. Sci, and J. B. Tenenbaum. Exploiting compositionality to explore a large space of model structures. In *Proc. UAI*, 2012.

[17] F. Han and S. Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE Trans. PAMI*, 31(1):59–73, 2009.

[18] M. Hofmann, M. Haag, and G. Rigoll. Unified hierachical multi-ojbect tracking using global data assoication. In *Proc. PETS Workshop*, 2013.

[19] M. Hofmann, D. Wolf, and G. Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *Proc. CVPR*, 2013.

[20] C. Huang, Y. Li, and R. Nevatia. Multiple target tracking by learning-based hierarchical association of detection responses. *IEEE Trans. PAMI*, 35(4):898–910, 2013.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding authors. In *Proc. ACM Multimedia*, 2014.

[22] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Trans. PAMI*, 31(2):319–336, 2009.

[23] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Sparse representation based image interpolation with nonlocal autoregressive modeling. *IEEE Trans. IP*, 22(4):1382–1394, 2013.

[24] S. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proc. ECCV*, 2006.

[25] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. Branch-and-price global optimization for multi-view multi-object tracking. In *Proc. CVPR*, 2012.

[26] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Proc. CVPR*, 2009.

[27] T. Liu, S. Chaudhuri, V. Kim, Q. Huang, N. Mitra, and T. Funkhouser. Creating consistent scene graphs using a probabilistic grammar. In *Proc. ACM SIGGRAPH*, 2014.

[28] X. Liu, L. Lin, and H. Jin. Contextualized trajectory parsing via spatio-temporal graph. *IEEE Trans. PAMI*, 35(12):3010–3024, 2013.

[29] X. Liu, L. Lin, S. Yan, and H. Jin. Adaptive tracking via learning hybrid template online. *IEEE Trans. CSVT*, 21(11):1588–1599, 2011.

[30] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *IEEE Trans. PAMI*, 33(11):2259–2272, 2011.

[31] L. Pero, J. Bowdish, E. Hartley, B. Kermgard, and K. Barnard. Understanding bayesian rooms using composite 3d object models. In *Proc. CVPR*, 2013.

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

[33] H. Possegger, T. Mauthner, P. Roth, and H. Bischof. Occlusion geodesics for online multi-object tracking. In *Proc. CVPR*, 2014.

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. NIPS*, 2015.

[35] X. Shi, H. Ling, J. Xing, and W. Hu. Multi-target tracking by rank-1 tensor approximation. In *Proc. CVPR*, 2013.

[36] H. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Multi-commodity network flow for tracking multiple people. *IEEE Trans. PAMI*, 36(8):1614–1627, 2013.

[37] X. Song, T. Wu, Y. Jia, and S.-C. Zhu. Discriminatively trained and-or tree models for object detection. In *Proc. CVPR*, 2013.

[38] A. Utasi and C. Benedek. A 3-d marked point process model for multi-view people detection. In *Proc. CVPR*, 2011.

[39] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *IJCV*, 32(9):184–204, 2013.

[40] B. Wang, G. Wang., K. L. Chan, and L. Wang. Tracklet association with online target-specific metric learning. In *Proc. CVPR*, 2014.

[41] L. Wen, W. Li, J. Yan, and Z. Lei. Multiple target tracking based on undirected hierarchical relation hypergraph. In *Proc. CVPR*, 2014.

[42] Z. Wu, N. Hristov, T. Hedrick, T. Kunz, and M. Betke. Tracking a large number of objects from multiple views. In *Proc. ICCV*, 2009.

[43] Y. Xu, L. Lin, W.-S. Zheng, and X. Liu. Human re-identification by matching compositional template with cluster sampling. In *Proc. ICCV*, 2013.

[44] Y. Xu, B. Ma, R. Huang, and L. Lin. Person search in a scene by jointly modeling people commonness and person uniqueness. In *Proc. ACMMM*, 2014.

[45] Y. Xu, H. Zhou, Q. Wang, and L. Lin. Realtime object-of-interest tracking by learning composite patch-based templates. In *Proc. ICIP*, 2012.

[46] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *Proc. CVPR*, 2007.

[47] A. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Proc. ECCV*, 2012.

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

[49] S. Zhang, X. Yu, Y. Sui, S. Zhao, and L. Zhang. object tracking with multi-view support vector machines. *IEEE Trans. MM*, 17(3):265–278, 2015.

[50] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem. Robust visual tracking via consistent low-rank sparse learning. *IJCV*, 111(2):171–190, 2015.

[51] Y. Zhao and S. Zhu. Image parsing via stochastic scene grammar. In *Proc. NIPS*, 2011.