Heterogeneous Association Graph Fusion for Target Association in Multiple Object Tracking

Hao Sheng, *Member, IEEE*, Yang Zhang, Jiahui Chen, Zhang Xiong, and Jun Zhang

***Abstract*— Tracking-by-detection is one of the most popular approaches to tracking multiple objects in which the detector plays an important role. Sometimes, detector failures caused by occlusions or various poses are unavoidable and lead to tracking failure. To cope with this problem, we construct a heterogeneous association graph that fuses high-level detections and low-level image evidence for target association. Compared with other methods using low-level information, our proposed heterogeneous association fusion (HAF) tracker is less sensitive to particular parameters and is easier to extend and implement. We use the fused association graph to build track trees for HAF and solve them by the multiple hypotheses tracking framework, which has been proven to be competitive by introducing efficient pruning strategies. In addition, the novel idea of adaptive weights is proposed to analyze the contribution between motion and appearance. We also evaluated our results on the MOT challenge benchmarks and achieved state-of-the-art results on the MOT Challenge 2017.**

***Index Terms*— Multiple object tracking, tracking-by-detection, target association, graph fusion.**

I. Introduction

TRACKING multiple targets from video sequences is the key technology in video understanding, motion recognition and event analysis, but it is still a vital challenge in computer vision. Though great improvement has been shown in the recent past, tracking in crowded and cluttered scenarios still has difficulties that need to be addressed, such as complex illumination variations, frequent occlusions and interactions among targets.

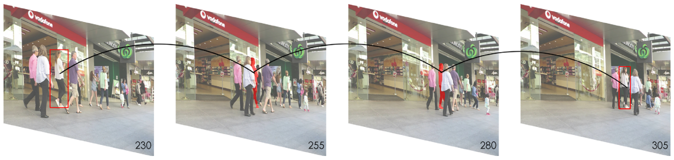


Fig. 1. Illustration of HAF tracker. The target in the red bounding box continues to be tracked by low-level image evidence when the detector fails. Black lines indicate the edges in our fused heterogeneous association graph. Only one target is marked out for clarity.

Tracking-by-detection is one of the most popular approaches in multi-target tracking due to the great progress on object detection. Targets of interest are extracted from the scene by a detector and then linked by different algorithms to form trajectories. In this way, the tracking task can be regarded as a data association problem. The tracker assigns each detection a unique ID corresponding to a certain target or discards it as a false alarm.

However, tracking targets when the detector fails is a tough challenge that can not be ignored. Detectors are designed to detect and localize objects in static images. However, detectors encounter a difficult problem when utilized to detect objects such as pedestrian which can vary greatly in appearance. Illumination variations, viewpoint or nonrigid deformations are some common factors of detection failure. For example, people of different genders wear different kinds of clothes or assume varieties of poses. Fortunately, the detector response is not the only information we can obtain from video sequences. It has been proven effective for tracking to utilize more image information, such as contour tracking [1], background modeling [2], superpixel segmentation [3] and etc. However, they have limitations in tracking targets under moving camera or in requiring manual initialization.

This paper proposes a heterogeneous association fusion method (HAF) to combine multi-layer information that simultaneously considers high-level target observations and low-level image evidence. A fused association graph is built to describe the association relationship between targets. In addition, a motion based segmentation algorithm is presented to extract foreground targets from whole scenes for both static and moving cameras. We build track trees with our fused association graph that can be solved by the multiple hypotheses tracking (MHT) framework. Compared with the amounts of parameters that need to be strategically tuned in SegTrack [3], we present a novel and easier method of using

Manuscript received May 30, 2018; revised September 5, 2018; accepted November 14, 2018. Date of publication November 19, 2018; date of current version October 29, 2019. This work was supported in part by the National Key R&D Program of China under Grant 2017YFC0803700, in part by the National Natural Science Foundation of China under Grant 61872025, in part by the Macao Science and Technology Development Fund under Grant 138/2016/A3, in part by the Open Fund of the State Key Laboratory of Software Development Environment under Grant SKLSDE-2017ZX-09, in part by the Project of Experimental Verification of the Basic Commonness and Key Technical Standards of the Industrial Internet Network Architecture, in part by the Technology Innovation Fund of China Electronic Technology Group Corporation, and in part by the HAWKEYE Group. This paper was recommended by Associate Editor Y. Wu. *(Corresponding author: Yang Zhang.)*

H. Sheng, Y. Zhang, J. Chen, and Z. Xiong are with the State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: shenghao@buaa.edu.cn; yang.zhang@buaa.edu.cn; chenjh@buaa.edu.cn; xiongz@buaa.edu.cn).

J. Zhang is with the Department of Electrical Engineering and Computer Science, University of Wisconsin–Milwaukee, Milwaukee, WI 53201 USA (e-mail: junzhang@uwm.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2018.2882192

superpixels by enhancing the MHT framework. Experimental results show our method is effective in tracking targets where the detector fails. Our main contributions are:

- An approach to build a heterogeneous association graph of multi-layer information including detector responses and image evidence to describe the relationship between observations;
- A HAF tracker with adaptive weights for motion and appearance;
- A method to extract foreground targets from the background in the superpixel level.

The rest of the paper is organized as follows. Related work is discussed in Sec.II. An approach to fuse a heterogeneous association graph with multi-layer information is presented in Sec.III. Exploiting heterogeneous association for tracking is described in Sec.IV. Experiment results are shown in Sec.V followed by the conclusion in Sec.VI.

II. Related Work

Multi-target tracking has been a popular topic in computer vision for years. Distinguishing targets from each other and contending with long-term occluded ones are two crucial problems that have gained much attention. Most recent tracking approaches can be generally categorized into two groups. One concentrates on the online processing technique, where the state posterior is estimated using only current and past observations. Kalman filters [4], [5] and particle filters [6], [7] are widely applied in real tracking applications. The former solves tracking recursively with two main steps, the prediction step and the correction step. The latter introduces a set of weighted particles sampled from a proposal distribution. Although they are limited to the increasing number of targets or Gaussian noise distributions, they have obvious advantages in simplicity and ease of implementation. However, there is a significant weakness in online methods in that they can not correct the trajectories when an early error is made.

In contrast, the entire sequence or a batch of the sequences is processed in offline tracking approaches. All frames are available in these methods. Tracking-by-detection is one of the most popular frameworks in recent research. Detections are generated by detectors in each frame independently and linked into trajectories. In this way, the multiple targets tracking task can be regarded as a data association problem and various methods are proposed.

Joint probabilistic data association (JPDA) [8] is an early algorithm for multi-target tracking. It computes the posterior probability of each targets in the validation gate that keeps the problem tractable. Multiple hypotheses tracking (MHT) is another conventional tracking method [9], [10]. It keeps a tree of hypotheses for each target and calculates the likelihood of branches to select the most likely one. Due to the crowded scenarios in the visual tracking field, pruning strategies are essential to address the exponential computational complexity. A recent work [11] presents that MHT approach remains the current suitable approach for advances in object detection and feature representation. They set dummy nodes to represent the case of missing observations which lead to failure in tracking long-term occluded targets.

More recently, linear programming based approaches are proposed for tracking. Jiang *et al.* [12] proposed a linear programming relaxation scheme that models tracking as a multi-path searching problem. Berclaz *et al.* [13] simplified the task by formulating the linking step as a constrained flow optimization and solved it using an efficient k-shortest paths algorithm. Zhang *et al.* [14] constructed a cost-flow network with a non-overlap constraint to solve data association and used the min-cost flow algorithm to find the optimal solution. Butt and Collins [15] used high-order motion information to introduce extra constraints and proposed an iterative solution method that makes the problem solvable by the min-cost flow algorithm. McLaughlin *et al.* [16] extended the min-cost network flow with a motion model to cope with long term occlusions and missed detections. Track interactions are modeled in [17] which combined different types of pairwise costs. Network flow based methods have the benefit of finding the globally optimal solution or approximate solution efficiently, but pairwise costs have limitations on incorporating with complex motion incorporation or the appearance model.

Milan *et al.* [18] proposed a conditional random field (CRF) model to joint data association and trajectory estimation. They formulated multi-target tracking as an energy minimization problem and described a minimization algorithm based on α-expansion, greedy label removal and continuous gradient based optimization. Solving data association to near global optimality and fitting trajectories to assigned detections are proceeded iteratively. However, lacking robustness in tracking occluded targets is a common shortcoming of all the methods mentioned above. To settle this issue, another CRF based method [3] was proposed to combine segmentation and tracking together by exploiting low-level image evidence. Both detections and superpixels are assigned with a unique target ID through sequences.

In fact, video segmentation has been applied to multi-target tracking much earlier. Bibby and Reid [1] built a level-set framework for visual tracking in real time. They modeled the discrete depth ordering of the objects and tracked the contours with occlusions, but required a manual initialization including the number of objects in the first frame. Mitzel *et al.* [19] presented an integrated framework that incorporates the contour of targets into the tracking-by-detection framework. However, a stereo camera rig is mounted for segmentation. In addition to tracking the contour of targets, background modeling is another popular approach for tracking [20]–[23]. The color of pixels are used to model the stationary background in order to extract foreground pixels as motion blobs. They proposed different algorithms to separate blobs into single ones. However, they failed to track targets when the camera is moving. Blobs can not be extracted because the background is changing through the frames.

In addition, some graph-based tracking approaches have been proposed in recent research. Wen *et al.* [24] incorporated multiple cameras to improve the robustness of tracking to occlusion and appearance ambiguities. They built a space-time-view hyper-graph to encode several constraints across different camera views. Zhang *et al.* [25] analyzed the start and the end location of tracking targets at time. To preserve

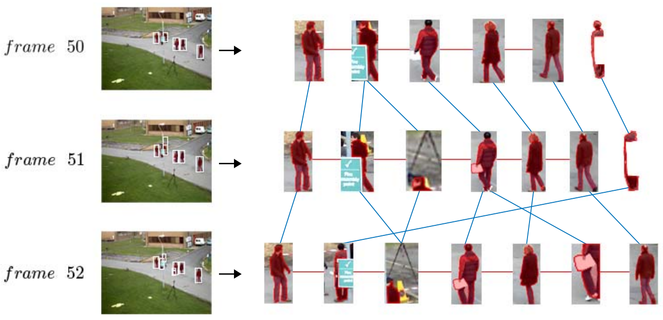


Fig. 2. Illustration of the fused heterogeneous association graph on PETS09-S2L1. Multi-layer information including high-level detections and low-level superpixels is considered to build the association graph. Detections are shown in white bounding boxes and foreground superpixels are marked with red areas. Temporal edges are shown in blue lines and spatial edges are shown in red lines. The rightmost nodes in F_{50} and F_{51} represent detection failure, so only foreground superpixels are shown. Only a subset of edges are shown for clarity.

the consistence of both the visual pattern and the spatial relationship, they formulated the task of tracklet association as a graph matching problem. Liu *et al.* [26] described a pipeline for multi-target visual tracking under a multi-camera system. They modeled tracking as a global graph and adopted generalized maximum multi-clique optimization. Both cross frame and cross camera data correlation were taken into account. However, in real-world applications, a multiple camera system may not be available considering the cost or environment.

The above mentioned studies do not present an approach with wide applicability to solve long-term detection failure in complex scenes. There are some limitations in these methods such as failure in moving scenes, requiring manual initialization or a calibrated multi-camera system, strong penalties of using low-level information, etc. In this paper, we aim to cope with the detection failure problem in different kinds of scenes.

III. Heterogeneous Association Graph Fusion

In this section, we describe the idea of solving detection failures by the graph fusion approach. Our work mainly concentrates on building the association between targets. When a detector fails due to partial occlusion or abnormal poses, a part of the low-level image evidence of the target is often still available. As a result, we construct a detection association graph for high-level detections and a superpixel association graph for low-level image evidence, and then fuse them together to represent the integrated association relationship between targets.

A. Detection Association Graph

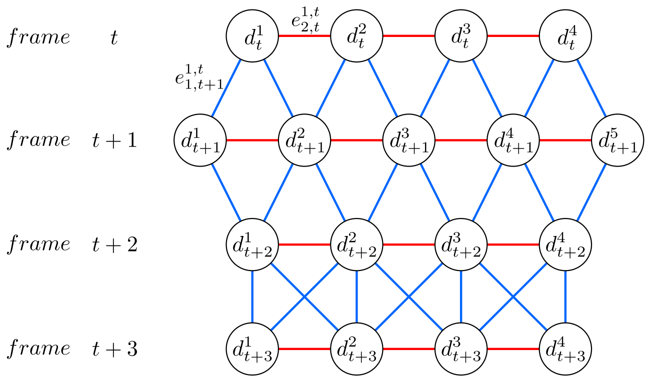


Fig. 3. Illustration of the detection association graph from F_t to F_{t+3}. Detections (V_{det}) are shown in the nodes. The red lines represent the spatial edges (E_{ds}) and the blue lines represent the temporal edges (E_{dt}). Only a subset of edges are shown for clarity.

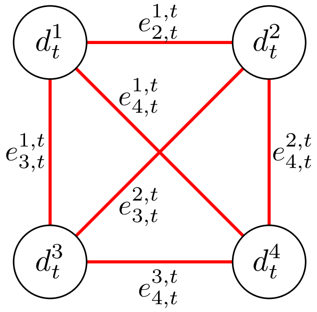


Fig. 4. Detections association in a single frame. Detections in the same frame are connected with spatial edges.

We follow tracking-by-detection approaches in this paper. Thus, building a realistic association between detections is the key for tracking, which means distinguishing targets from one another and constructing trajectories to explain the targets' motion correctly in real world scenes. From this point of view, we construct a detection association graph to describe the relationship between detections. We obtain detections as bounding boxes from an object detector similar to most other methods. Let F_t denotes the t^{th} frame of the video sequence and we use $\{d_t^1, d_t^2, \ldots, d_t^k\}$ to define k detections in F_t.

We define the detection association graph as follows:

$$G_{det} = \{V_{det}, E_{dt}, E_{ds}\} \tag{1}$$

where V_{det} is the set of all detections, E_{dt} represents the set of temporal edges that link detections between neighboring frames, and E_{ds} represents the set of spatial edges that link detections in the same frame. These two kinds of edges are defined for different usage. The temporal edges are used to build trajectories of potential targets. They describe the likelihood between observations in adjacent frames in terms of motion and appearance. On the other hand, the spatial edges make contributions to analyzing the discrimination between observations in the same frame, which will be make further discussed later.

Detections in the same frame are connected by spatial edges, thus d_t^1 and d_t^2 are connected by $e_{2,t}^{1,t} \in E_{ds}$. Temporal edges link detections in the neighboring frames, d_t^1 and d_{t+1}^1 are connected by $e_{1,t+1}^{1,t} \in E_{dt}$ for example. For a given frame, detections are fully connected as shown in Fig. 4.

The cost of spatial edges C_{ds} is defined using appearance features of detections, while the cost of temporal edges C_{dt} considers both motion and appearance features. The detailed definitions are described in Sec.IV-B and Sec.IV-C.

B. Superpixel Association Graph

Although object detectors have made remarkable progress [27]–[29], they still unavoidably fail when targets are partially or totally occluded. However, other information in addition to detection responses such as pixel-level features, is still available for the case where only a part of the target is invisible. Such image evidence provides clues for tracking to some extent. Our goal is to keep targets associated even though the detector fails. We present a method to construct an association graph between targets with low-level information. To reduce the complexity and gain convenient image features [30], we use superpixel-level features instead of pixel-level features for our association graph.

It is necessary to extract the foreground from the background before constructing an association graph with superpixels. Although there are numerous state-of-the-art algorithms [31]–[33] in the background modeling area, they mainly focus on extracting the foreground in static scenes. In a tracking problem, when videos are taken by moving cameras such as hand-held cameras, these foreground extraction algorithms fail to provide acceptable results. To gain foreground observations for tracking, Milan *et. al* [3] proposed a linear SVM segmentation method with color features. However, we find that amounts of superpixels are labeled improperly when the foreground and background have a low degree of differentiation in color space. To cope with this issue, we present an approach to extract foreground pixels in moving scenes (Alg.1).

Note that in most cases, there is relative movement between foreground observations and the background. We use optical flow features instead of color for modeling. A quadric surface is fitted as the background for every frame by the optical flow of pixels outside of detections. Then, for each frame, all pixels P are sorted according to the absolute difference to the fitted surface. A pixel is selected as belonging to the foreground pixels P_f if its absolute difference is higher than the average.

We obtain temporal superpixels by employing the method in [34], and the foreground likelihood $\mathcal{F}_i$ of each superpixel s_i is defined as the percentage of foreground pixels in each superpixel. Superpixels that have $\mathcal{F}_i$ over threshold $\mathcal{F}_{th}$ are labeled as foreground superpixels.

$$\mathcal{F}_i = \frac{\sum_{p_j^f \in s_i} p_j^f}{\sum_{p_j \in s_i} p_j} \tag{2}$$

Based on foreground superpixel extraction, we define a *superpixel set* (in italic) as a set of maximally connected foreground superpixels. All superpixels in this set belong to the same detection or they are located outside of any detections. Let $\{s_t^1, s_t^2, \ldots, s_t^k\}$ represents k *superpixel sets* in F_t.

we define the superpixel association graph as follows:

$$G_{sp} = \{V_{sp}, E_{st}\} \tag{3}$$

where V_{sp} denotes the set of all *superpixel sets* and E_{st} denotes the temporal edges that link *superpixel sets* between neighboring frames. Different from the detection association graph, we do not build spatial edges in the superpixel association graph. It is because that the spatial information has already been used when generating *superpixel sets*. Temporal edges in the superpixel association graph describe the likelihood of superpixel sets by considering both motion and color features.

Algorithm 1 Foreground Pixels Extraction

```
Input: Detections $D$, Optical Flows $F$, Image Size $(w, h)$
Output: Foreground pixels $P_f$
for each frame $i \in [1, t]$ do

    $d = D$ in $i$ //detections in frame i
    $f = F$ in $i$ //optical flow of entire image in frame i

    //optical flow of pixels outside of detections
    $f_{bg} = f$ outside $d$

    //fit quadric surface
    $background$ = fitPoly2($f_{bg}$)

    $diff$ = abs($background - f$)

    //calculate the average difference
    $ave$ = sum($diff$) / $(w * h)$

    $p_i = diff > ave$ ? 1 : 0
end for
$P_f = \{p_1, p_2, \ldots, p_t\}$                ▷ $p_i = 0$ or 1
return $P_f$
```

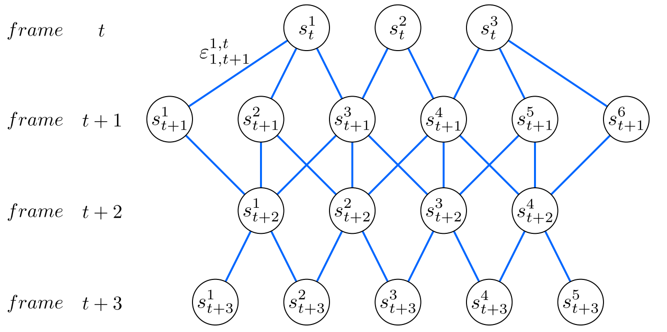


Fig. 5. Illustration of superpixel association graph from F_t to F_{t+3}. *Superpixel sets* (V_{sp}) are shown in nodes. The blue lines indicate the temporal edges (E_{st}). Only a part of the edges are shown to keep the figure readable.

As shown in Fig. 5, node s_t^i represents the i^{th} *superpixel set* in F_t. Temporal edges that link *superpixel sets* in the neighboring frames, such as s_t^1 and s_{t+1}^1 are connected by $\varepsilon_{1,t+1}^{1,t} \in E_{st}$. The cost of temporal edges C_{st} is obtained by the temporal label and is specified in Sec.IV-B.

C. Association Graph Fusion

It is obvious that our proposed detection association graph and superpixel association graph are heterogeneous and non-isomorphic as well. To avoid ambiguity, we define a

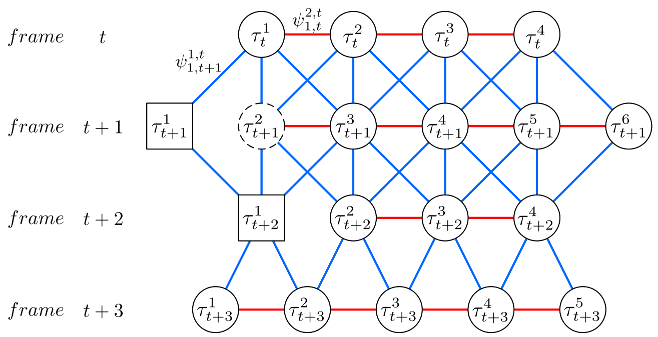


Fig. 6. Illustration of the fused association graph from F_t to F_{t+3}. *Observations* (V) are shown in three kinds of nodes, circle for $\mathcal{DS}$, dashed circle for $\mathcal{D}$ and square for $\mathcal{S}$. Red lines represent spatial edges. Blue lines represent temporal edges. Only a part of the edges is shown for clarity.

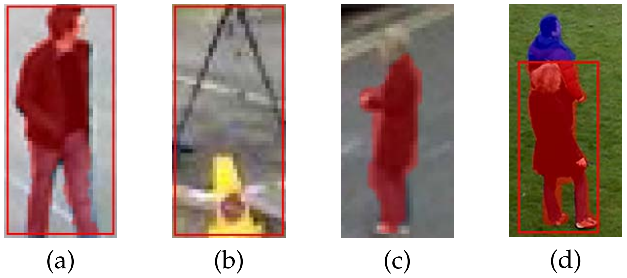


Fig. 7. Three kinds of *observations* in PET09-S2L1. Detections are shown in red bounding boxes and *superpixel sets* are illustrated as semitransparent masks. (d) shows the case when a *superpixel set* intersects with a detection. It is split into two *superpixel sets* (red and blue). As a result, there is a DS and an S in (d).

unified description that an *observation* (in italic) means a detection or a *superpixel set*. Our goal is to fuse the detection association graph and superpixel association graph to describe the targets' association relationship more robustly. The fused association graph is defined as:

$$G = \{V, E_t, E_s\} \tag{4}$$

where V is the set of all *observations* generated from V_{det} and V_{sp}. E_t is the set of temporal edges linking *observations* between neighboring frames and E_s represents the set of spatial edges linking detections in the same frame. Let τ_t^i denotes the i^{th} *observation* in F_t. Thus, V can be expressed as:

$$V = \{\tau_1^1, \tau_1^2, \ldots, \tau_t^i\} \tag{5}$$

As shown in Fig. 6 and Fig. 7, there are three kinds of *observations* in the fused graph: detections containing the *superpixel set* (called $\mathcal{DS}$), detections containing no *superpixel sets* (called $\mathcal{D}$), *superpixel sets* not belonging to any detections (called $\mathcal{S}$). Specifically, if a *superpixel set* intersects with a detection and partially belongs to detections, it will be split into multiple *superpixel sets* according to the bounding boxes. E_t is a super set of E_{dt} and E_{st} that links *observations* between neighboring frames, τ_t^1 and τ_{t+1}^1 are linked by $\psi_{1,t+1}^{1,t} \in E_t$ for example. In addition, E_s links *observations* generated by detections ($\mathcal{D}$ and $\mathcal{DS}$) in the same frame, such as $\psi_{1,t}^{2,t} \in E_s$.

$$\begin{aligned} V = &\{d_t^i | \exists s_t^j \in V_{sp}, s_t^j \in d_t^i\} \\ &\cup \{d_t^i | \forall s_t^j \in V_{sp}, s_t^j \notin d_t^i\} \\ &\cup \{s_t^i | \forall d_t^j \in V_{det}, s_t^i \notin d_t^j\} \end{aligned} \tag{6}$$

The costs of temporal edges C_t and spatial edges C_s in the fused association graph are defined as follows:

$$C_t = \omega_{det} C_{dt} + \omega_{sp} C_{st} \tag{7}$$

$$C_s = C_{ds} \tag{8}$$

Note that there are nine kinds of temporal edges among *observations*, $C_t(\mathcal{DS}, \mathcal{DS})$, $C_t(\mathcal{DS}, \mathcal{D})$, $C_t(\mathcal{DS}, \mathcal{S})$, $C_t(\mathcal{D}, \mathcal{DS})$, $C_t(\mathcal{D}, \mathcal{D})$ $C_t(\mathcal{D}, \mathcal{S})$, $C_t(\mathcal{S}, \mathcal{DS})$ $C_t(\mathcal{S}, \mathcal{D})$ and $C_t(\mathcal{S}, \mathcal{S})$. They are defined in detail in Sec.IV-B.

IV. Heterogeneous Association Fusion in Multiple Hypotheses Tracking

To apply heterogeneous association fusion for tracking, we adopt the multiple hypotheses tracking (MHT) framework. Based on the fused association graph, temporal edges are used for building track trees that includes both detection nodes and superpixel nodes. In addition, we propose adaptive weights for tracking scores by use of spatial edges.

A. Track Tree Construction

Each track tree represents the track hypotheses of a target. Similar to [11], we also build new trees and extend existing trees at each frame. However, we use the *observation* (defined in Sec.III-C) as the node of the tree instead of only detections. In this way, both high-level detector responses and low-level image evidence are taken into consideration for tracking.

For a given frame, we build new trees for each *observation* in the fused association graph in that frame, indicating if a new trajectory appears. Existing trees are updated with *observations* from the frame as well. Each tree is extended by appending *observations* as its children to build a branch. We also adopt the dummy node strategy for missing detections in case of both detections and superpixels are invalid for tracking. Dummy nodes are used to account for missing *observations*. Branches are deleted from the trees if they have N_{miss} consecutive dummy nodes.

To control the scale of trees, a filter is applied to decide whether to update the tree with the *observation* or not. Let $\hat{\tau}_t^i$ denote the prediction of τ_{t-1}^j in the current frame. Its locations is defined as the center locations of all pixels located in the detection's bounding box or the *superpixel set* (determined by whether τ_t^i is $\mathcal{DS}$, $\mathcal{D}$ or $\mathcal{S}$). We use optical flow to predict $\hat{\tau}_t^i$. Let f_{t-1}^j represents the optical flow of τ_{t-1}^j, defined as the mean optical flow of all superpixels located in the τ_{t-1}^j. The distance d_τ between τ_t^i and $\hat{\tau}_t^i$ is defined based on their Euclidean distance:

$$d_\tau^2(\tau_t^i, \hat{\tau}_t^i) = (\tau_t^i - \hat{\tau}_t^i)^T (\tau_t^i - \hat{\tau}_t^i) \tag{9}$$

$$\hat{\tau}_t^i = \tau_{t-1}^i + f_{t-1}^i \tag{10}$$

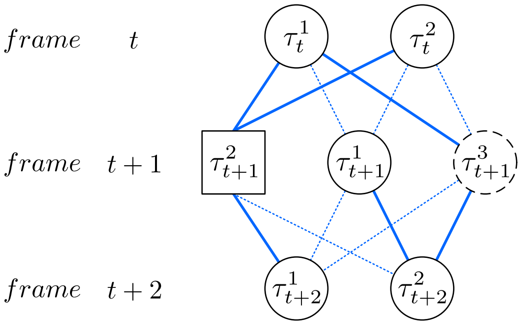


Fig. 8. To clarify the construction of track tree, we show a brief instance of the fused association graph and only temporal edges are shown as spatial edges are not used for track tree construction. Dashed lines represent that the distance between *observations* is over d_{th}.

Observations far from the predicted location (over the threshold d_{th}, $d_\tau(\tau_t^i, \hat{\tau}_t^i) > d_{th}$) are not used for updating the tree.

An instance of the track tree construction is shown in Fig. 9. Each branch has a score determined by the summation of the corresponding temporal cost in the fused association graph. Let b_k represent the k^{th} branch in the track trees. The score for b_k is defined as:

$$S_k = \sum_{(\tau_t^i, \tau_{t+1}^j) \in b_k} C_t(\tau_t^i, \tau_{t+1}^j) \tag{11}$$

Our proposed track tree is built according to the fused heterogeneous graph. Thus, in contrast to track trees in other approaches [10], [11], there are detection nodes and superpixel nodes in the trees, which can maintain a stronger association between targets than only using dummy nodes when the detector fails.

B. Temporal Edges

The cost of temporal edges in the fused association graph consists of two parts, C_{dt} and C_{st}. We first describe the former part, the cost of temporal edges in the detection association graph. It is defined as follows:

$$C_{dt} = \omega_{mot} C_{mot} + \omega_{app} C_{app} \tag{12}$$

where ω_{mot} and ω_{app} control the weights of motion and appearance costs of detections. We will further discuss these two weights in Sec.IV-C by using spatial edges.

We use the optical flow feature to evaluate the motion likelihood. Let $\hat{d}_{t+1}^i$ denotes the prediction of d_t^i in F_{t+1}, and the width and height of $\hat{d}_{t+1}^i$ are assigned as same as the d_t^i. The motion cost between d_t^i and d_{t+1}^j is defined as:

$$C_{mot}(d_t^i, d_{t+1}^j) = \frac{\hat{d}_{t+1}^i \cap d_{t+1}^j}{\hat{d}_{t+1}^i \cup d_{t+1}^j} \tag{13}$$

$$\hat{d}_{t+1}^i = d_t^i + f_t^i \tag{14}$$

where f_t^i is the mean optical flow of all foreground superpixels in d_t^i. The edges between detections have higher C_{mot} (higher overlap rate) when the spatial relationship between detections can reasonably describe their movement in the real world.

As for the appearance likelihood, we utilize the convolutional neural network features from GoogLeNet [35] and extract 256-dimensional feature for each detection. Let a_t^i denotes the appearance feature vector of d_t^i. We define the cosine distance between detections as the appearance cost:

$$C_{app}(d_t^i, d_{t+1}^j) = \frac{a_t^i \cdot a_{t+1}^j}{\left\| a_t^i \right\| \left\| a_{t+1}^j \right\|} \tag{15}$$

The value of C_{app} ranges from -1 to 1, representing the appearance likelihood between detections from least to most. Higher C_{app} means that two detections of the edge are more likely to be the same target.

We now discuss the cost of temporal edges in the superpixel association graph. As the superpixels are obtained by [34], an identity label is assigned to each superpixel. Superpixels in the same frame have different labels but share the same label between adjacent frames if they are regarded as the same object. These temporal labels are not convincing in a long period of time, but they have high confidence between two neighboring frames. Hence we use these labels to define the temporal cost between *superpixel sets*. Let $L_t^i = \{l_1, l_2, \ldots, l_m\}$ represents the labels of superpixels of s_t^i and $L_{t+1}^j = \{l_1, l_2, \ldots, l_n\}$ represents the labels of superpixels of s_{t+1}^j, where m and n are the number of superpixels of s_t^i and s_{t+1}^j. The cost of temporal edges in the superpixel association graph is defined as:

$$C_{st}(s_t^i, s_{t+1}^j) = \frac{|L_t^i \cap L_{t+1}^j|}{max(m, n)} \tag{16}$$

As discussed in Sec.III-C, there are nine kinds of temporal edges in the fused association graph among *observations*. They are generally defined in a unified form as Eq.(7).

However, the nodes in the fused association graph are heterogeneous, including $\mathcal{DS}$, $\mathcal{D}$ and $\mathcal{S}$. According to our definition, there are no foreground superpixels in $\mathcal{D}$ and $\mathcal{S}$ does not belong to any detections, i.e., C_{st} or C_{dt} may not exist. As a result, we use the following method for these situations. The cost of temporal edges in the fused association graph can be expressed as:

$$\begin{aligned} C_t(\mathcal{DS}, \mathcal{DS}) &= \omega_{det} C_{dt} + \omega_{sp} C_{st} \\ C_t(\mathcal{DS}, \mathcal{D}) = C_t(\mathcal{D}, \mathcal{DS}) &= \omega_{det} C_{dt} \\ C_t(\mathcal{DS}, \mathcal{S}) = C_t(\mathcal{S}, \mathcal{DS}) &= \omega_{det} C'_{dt} + \omega_{sp} C_{st} \\ C_t(\mathcal{D}, \mathcal{D}) &= \omega_{det} C_{dt} \\ C_t(\mathcal{D}, \mathcal{S}) = C_t(\mathcal{S}, \mathcal{D}) &= \omega_{det} C'_{dt} \\ C_t(\mathcal{S}, \mathcal{S}) &= \omega_{sp} C_{st} \end{aligned} \tag{17}$$

where ω_{det} is the mean value of the detections' confidence (from 0 to 1), and ω_{sp} is the mean value of the superpixels' foreground score. In the case of $C_t(\mathcal{DS}, \mathcal{S})$, $C_t(\mathcal{D}, \mathcal{S})$, $C_t(\mathcal{S}, \mathcal{DS})$ and $C_t(\mathcal{S}, \mathcal{D})$, let $\hat{d}_{t\pm1}^i$ represent the prediction of d_t^i with the same width and height, but assign 0 as its confidence. Let $\hat{s}_{t\pm1}^i$ denote the subset of $s_{t\pm1}^i$ located in the

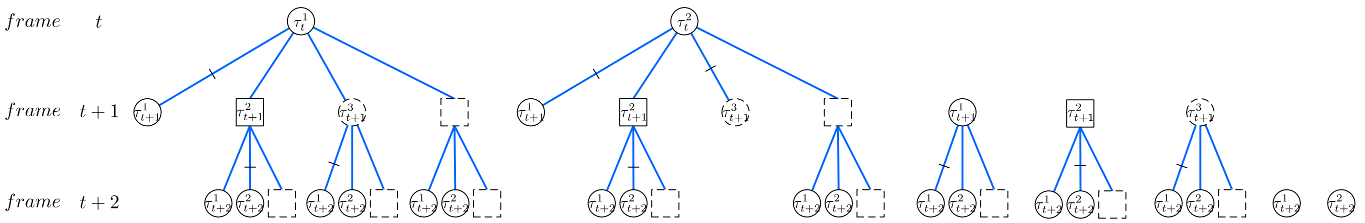


Fig. 9. Illustration of the track trees constructed by Fig. 8. Existing trees are updated and new trees are built in each frame. Circle for $\mathcal{DS}$, dashed circle for $\mathcal{D}$, square for $\mathcal{S}$ and dashed square for dummy node. Branches are cut if the distance between *observations* is over the threshold as shown in dashed lines in Fig. 8.

$\hat{d}^i_{t\pm1}$. We define C'_{dt} as:

$$C'_{dt} = \frac{|\hat{s}^i_{t\pm1}|}{|s^i_{t\pm1}|} \tag{18}$$

$$\hat{d}^i_{t\pm1} = d^i_t \pm f^i_t \tag{19}$$

where f^i_t is the forward (or backward) mean optical flow of d^i_t. Furthermore, the cost of temporal edges linking at least one dummy node is set to 0 as a penalty.

C. Spatial Edges

Motion and appearance features are widely used for tracking to obtain plausible trajectories. Some methods build motion model with severe restrictions, such as constant velocity [36] or linear motion [16]. These constraints are practical and efficient in sparse crowd scenarios and can generate smooth trajectories. Unfortunately, when the camera is moving or the target suddenly changes its velocity, the strong restrictions on motion often lead to poor results. Some recent methods take appearance information into account to improve the accuracy of tracking. However, targets have little differences on appearance in some cases, so it is not reasonable to keep excessive weight on appearance, $\omega_{app}/\omega_{mot} = 9$ in MHT_DAM [11] for example.

It is worthwhile to study the contributions from motion and appearance. Based on the spatial edges in the fused association graph, we reconsider the balance between motion and appearance and propose adaptive weights by using spatial edges. In our HAF tracker, ω_{mot} and ω_{app} in Eq.(12) are adaptively determined by the changing discrimination on the appearance through the frames. We define the cost of spatial edges as the cosine distance between the detections' appearance:

$$C_s(\tau^i_t, \tau^j_t) = C_{ds}(d^i_t, d^j_t) = \frac{a^i_t \cdot a^j_t}{\|a^i_t\| \|a^j_t\|} \tag{20}$$

We use the standard deviation of all C_s in each frame to define ω_{mot} and ω_{app}. Let $\omega_{app}(t)$ and $\omega_{mot}(t)$ represent the weight of appearance and motion information in F_t. Thus they are defined as:

$$\omega_{app}(t) = \sqrt{\frac{1}{N}\sum_{1}^{N}(C_s(\tau^i_t, \tau^j_t) - \mu)^2} \tag{21}$$

$$\omega_{mot}(t) = 1 - \omega_{app}(t) \tag{22}$$

where N is the number of spatial edges in F_t and μ is the mean value of C_s in F_t.

D. Global Hypotheses Optimization

We can obtain a set of trajectory hypotheses for all targets from the track trees after scoring and pruning, but there is more than one hypothesis for each target. It is because new track trees are built for every target in each frame, and all track trees are pruned independently. As a result, an *observation* may exist in many hypotheses.

To ensure that each *observation* (including DS, D and S) is assigned to a unique trajectory, we follow the idea in [11] to formulate this task as the following k-dimensional assignment problem. This problem is NP-hard when k is greater than 2. As a result, it can be formulated as a maximum weighted independent set (MWIS) problem to find the most likely set of trajectories according to their score calculated by Eq.(11).

We also utilize the exact algorithm [37] or an approximate algorithm [38] to solve the MWIS problem depending on its level of difficulty.

V. Experiments

In this section we discuss the parameters used in the experiments, and then show qualitative and quantitative tracking results on public benchmarks.

A. Datasets and Metrics

We test our tracking method on both the MOT Challenge 2016 [39] and 2017. They are widely used for a fair comparison in recent years. Both of them contain video sequences in unconstrained environments filmed with both static and moving cameras. There are 14 sequences (7 training, 7 test) with 11,235 frames in MOT Challenge 2016, and 42 sequences (21 training, 21 test) with 33,705 frames in MOT Challenge 2017. For a fair comparison with other tracking approaches, we use the publicly available detections provided by MOT Challenge 2016 and 2017.

For evaluation, we adopt the widely used CLEAR MOT metrics [40]. MOTA↑ (multiple object tracking accuracy) combines three kinds of errors including FP↓ (false positives), FN↓ (false negatives) and IDS↓ (identity switches). MOTP↑ (multiple object tracking precision) is another score to show the precision of the output trajectories against ground truth. IDF1 [41] is the ratio of correctly identified detections over the average number of ground truth and computed detections. Additionally, MT↑ (mostly tracked, > 80%), ML↓ (mostly lost, < 20%), track fragmentations (FM)↓ and Hz↑ (processing speed, frames per second) are also reported.

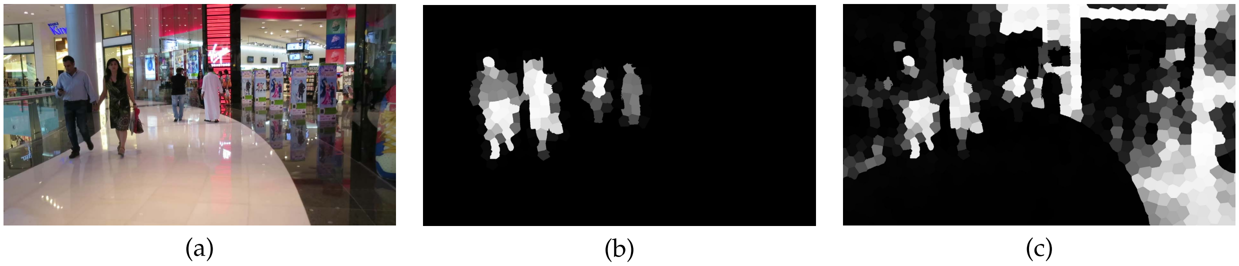

(a) (b) (c)

Fig. 10. Exemplar frames of segmentation results from MOT16-12 (frame 33). (a) shows the images from the sequences. (b) is the result of our segmentation results and (c) shows the results in [3]. A brighter superpixel means a higher foreground likelihood.

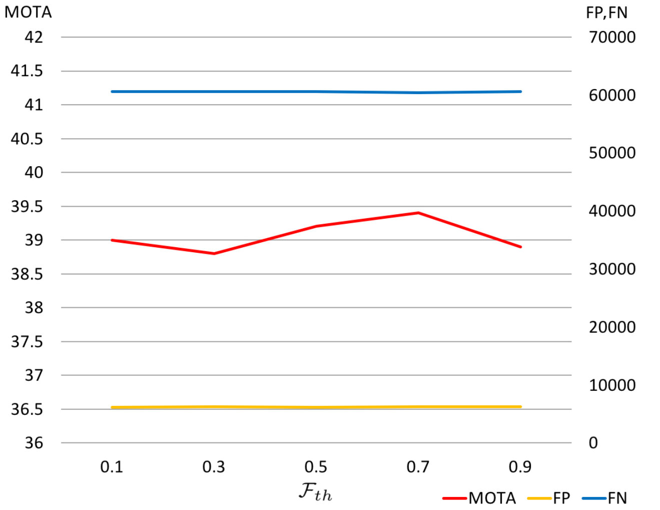


Fig. 11. Tracking results with different $\mathcal{F}_{th}$ (set as 0.1, 0.3, 0.5, 0.7 and 0.9) on the MOT Challenge 2016 training set. MOTA, FP and FN are indicated in the figures.

The indicator ↑ means the higher the better and ↓ means the lower the better.

B. Parameters and Robustness

In Sec.III, we proposed a foreground extraction algorithm and a threshold $\mathcal{F}_{th}$ is set to determine whether a superpixel belongs to foreground or background. We present several experiments to show its setting. As shown in Fig. 11, we set different $\mathcal{F}_{th}$ as 0.1, 0.3, 0.5, 0.7, 0.9, and the results show that it is not a sensitive parameter. We fix the $\mathcal{F}_{th}$ to 0.5, which is the setting for all other experiments with moving cameras in this paper. For scenes taken by static cameras, we use a mature background modeling algorithm (ViBe [42]) to obtain foreground pixels.

Qualitative segmentation results (moving camera) are shown in Fig. 10, brighter superpixels means they have higher foreground likelihood. There is a noticeable phenomenon that the segmentation algorithm in SegTrack [3] has a preference to label superpixels in dark color as the foreground. It can be seen that almost all black areas have a high foreground likelihood in Fig. 10-(c). It is because their positive and negative training samples are obtained by color clustering and most positive samples are from the clothes. It leads to a high sensitivity to color. However, there are more people wearing dark clothing than other colors in the video sequences. As a result, in Fig. 10-(c), the man in the center with white clothes is labeled as background, and only the black pants (the man on the left with light blue shirt) and the black shirt (the man in the center with light blue jeans) are labeled as foreground. In Fig. 10-(b), we obtain a more accurate segmentation results in such scenes by using motion features that are not sensitive to color.

There are two important parameters in MHT based trackers for pruning. One is the N-scan pruning parameter N, the other one is the maximum number of branches B_{th}. Compared with the traditional MHT framework, our HAF tracker extends the track trees with superpixel nodes in order to track more targets, so it is necessary to analyze whether the original pruning parameters are still appropriate in our method.

The tracking results with different N-scan parameters are shown in Fig. 12. In this paper, we propose a novel score for pruning that consists of motion and appearance information of detections and the association relationship between superpixels. We set different N from 1 to 10 while the B_{th} is fixed to 500 which is large enough to show the effect by N independently. The results show that HAF tracker is not sensitive to the N change. Even when N is set to a small value, false tracking hypotheses can still be removed and MOTA is impressive. It proves that our proposed cost function for pruning is rational and effective.

Another important parameter is B_{th}, which controls the maximum number of branches. There are extra superpixel nodes in HAF, leading to a larger scale of track trees. Therefore, a larger B_{th} may be needed to keep correct hypotheses. We perform a set of experiment to observe the sensitivity of B_{th} from 50 to 150. In Fig. 13, the results are fairly smooth when B_{th} changes from 70 to 150. It provides strong justification for the robustness of superpixel nodes on retaining possible hypotheses.

These results demonstrate that HAF tracker reserves the advantages of MHT_DAM [11]. It means that our cost function and superpixels nodes do not reduce the robustness to different pruning parameter settings. In the following experiments, we fix $N = 6$, $B_{th} = 100$, $N_{miss} = 15$ (consecutive dummy nodes) and $d_{th} = 12$ in Eq.(9) as the same as MHT_DAM [11] for fair comparisons.

C. Dummy Nodes Analysis

In the MHT framework, a separate branch with a dummy observation is grown to indicate a missing detection when a

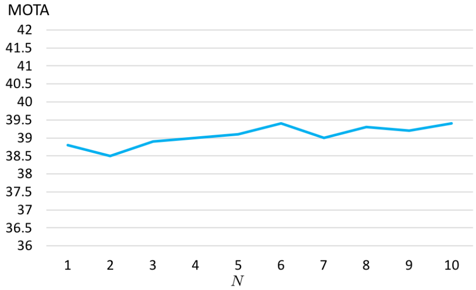


Fig. 12. Tracking results with different N (set from 1 to 10) on the MOT Challenge 2016 training set. B_{th} is fixed to 500. MOTA is indicated in the figures.

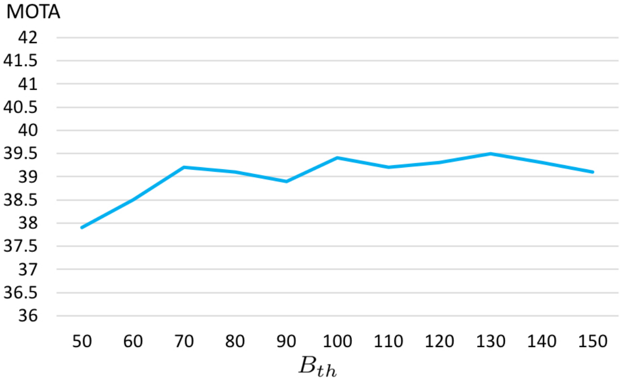


Fig. 13. Tracking results with different B_{th} (set from 50 to 150) on the MOT Challenge 2016 training set. N is fixed to 6. MOTA is indicated in the figures.

TABLE I

Dummy Nodes and Speed (MOT 2016 Training)

Sequence	Baseline [11]	Ours
MOT16-02	215,919 (0.5)	282,744 (0.5)
MOT16-04	950,581 (0.2)	1,154,510 (0.2)
MOT16-05	130,666 (1.0)	142,871 (1.0)
MOT16-09	190,189 (0.3)	243,097 (0.3)
MOT16-10	227,532 (0.6)	304,586 (0.6)
MOT16-11	219,964 (0.7)	284,997 (0.6)
MOT16-13	122,347 (1.1)	150,050 (1.2)
Total	2,057,198 (0.6)	2,562,855 (0.6)

hypothesis is extended by a new observation. HAF tracker extends hypotheses not only by detections but also by superpixels, so there are undoubtedly more dummy nodes in track trees. We count the number of dummy nodes in both MHT_DAM and HAF. The results in Tab. I show that our method has 505,657 more dummy nodes than the baseline and the speed is shown in the parentheses (Hz). Although there are approximately 24.5% more nodes in HAF, the speed of the tracker is mainly determined by the pruning parameters and the global hypotheses optimization algorithm. As discussed earlier, we use the same parameters and optimization algorithm as MHT_DAM, so the extra nodes in HAF do not reduce the speed of the tracker.

D. Adaptive Weights

The spatial edges in the heterogeneous association graph are used for adaptive weights as discussed in Sec.IV-C. It can evaluate the contributions between motion constraints and appearance features through frames. To prove the effectiveness of adaptive weights, we test our idea on the MOT Challenge 2016 Training set by setting different weights on motion and appearance factors. We set up 5 contrast experiments as shown in Tab. II. The weights of the baseline tracker are set to 0.1-0.9 according to [11]. In addition to one with adaptive weight, the others are set as 0.1-0.9, 0.5-0.5 and 0.9-0.1.

In Tab. II, HAF achieves the lowest MOTA score when setting the weight too large on the motion factor. Compared with the appearance, the motion features are more likely to change suddenly due to an unpredictable motion trend. As a result, setting a larger weight on the appearance can reduce FP and improve the MOTA score. However, in MHT based tracking approaches, the tracking task is converted to finding a branch in the track tree that best explains the trajectory in real scenes. The key of finding the correct branch is to give each branch a reasonable score, and the weights of motion and appearance have a great influence on the score. When the weights are fixed, the variation in a video can not be reflected over time, and the difference among videos is also ignored. The one with adaptive weights achieves the highest MOTA compared to the others, and it is a demonstration that our method has the ability to express the varying contributions between the motion and appearance in different frames.

E. Benchmark Comparison

As MHT_DAM [11] is a similar MHT based tracker to our HAF tracker, and they submitted their results on both MOT Challenge 2016 and 2017. We regard MHT_DAM as a baseline to evaluate our HAF tracker and compare with other state-of-the-art trackers as well. Trackers are performed on different platforms, so the speed is simply a reference.

Tab. III shows our experimental results (denoted as eHAF16) on the MOT Challenge 2016, and the best two results of each metric are shown in bold. MOTA and IDF1 are two aggregative metrics to evaluate the performance of trackers. The proposed eHAF16 tracker takes the third place sorted by MOTA score (47.2) and the second place sorted by IDF1 (52.4). We achieve the highest MT (18.6%) and second lowest FN (83,107), which shows the validity of our original intention. The extra superpixel nodes have a strong ability to reduce FN and thus to keep possible hypotheses. Compared with MHT_DAM, eHAF16 outperforms it by 1.4 on MOTA and 6.3 on IDF1. Although there are 6,174 more FP than MHT_DAM, FN is reduced by 8,651. MT is also improved from 16.2% to 18.6% which proves our method is effective to recover more trajectories of targets.

In the more recent MOT Challenge 2017, our results are shown in Tab. IV. The best result of each metric is shown in bold. Our tracker shows state-of-the-art performance by achieving the best score on most metrics including MOTA, IDF1, MT, FN, IDS and FM. Compared with MHT_DAM, we outperform it on MOTA by 1.1, IDF1 by 7.5, MT by 2.6%, FN by 16,117, IDS by 471 and FM by 126.

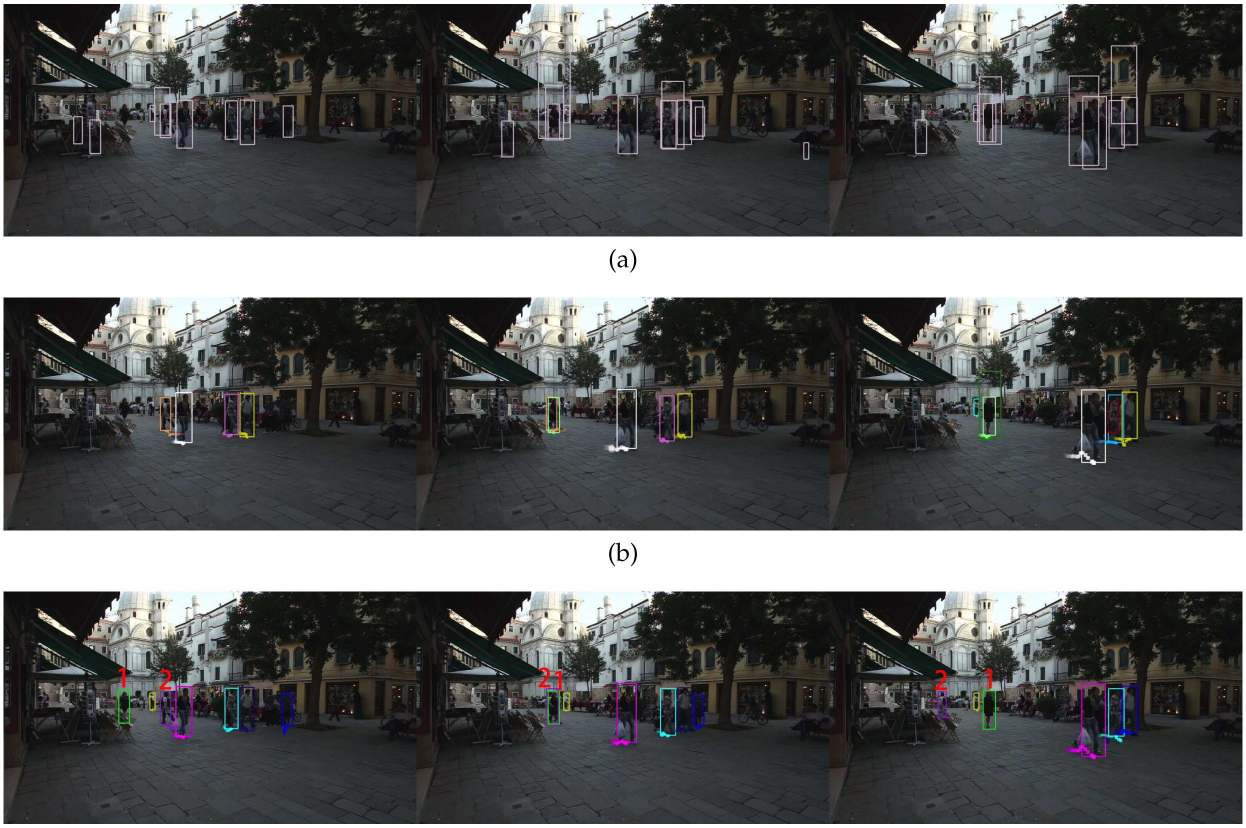
In both benchmarks, FNs are lowered as expected and FPs increase in an acceptable range. Our tracker considers both

TABLE II
ADAPTIVE WEIGHTS EFFECTIVENESS (MOT 2016 TRAINING)

Method (motion, appearance)	MT	ML	FP	FN	IDs	FM	MOTA	MOTP
Baseline [11] (0.1, 0.9)	62	284	**3,014**	68,044	**228**	**312**	35.4	**78.7**
Ours (0.1, 0.9)	**88**	**230**	7,347	60,009	334	483	38.7	77.7
Ours (0.5, 0.5)	**88**	235	7,456	60,075	329	485	38.5	77.7
Ours (0.9, 0.1)	84	232	8,653	**59,957**	331	502	37.6	77.7
Ours (adaptive weights)	**88**	237	6,187	60,580	310	455	**39.2**	77.8

TABLE III
RESULTS ON MOT CHALLENGE 2016 TEST(5/1/2018)

Method	MOTA	IDF1	MT	ML	FP	FN	IDS	FM	Hz
LMP [43]	**48.8**	51.3	18.2%	40.1%	6,654	**86,254**	481	595	0.5
NLLMPa [44]	**47.6**	47.3	17.0%	40.4%	5,844	89,093	629	768	**8.3**
eHAF16 (ours)	47.2	**52.4**	**18.6%**	42.8%	12,586	**83,107**	542	787	0.5
AMIR [45]	47.2	46.3	14.0%	41.6%	**2,681**	92,856	774	1,675	1.0
NOMT [46]	46.4	**53.3**	**18.3%**	41.4%	9,753	87,565	**359**	**504**	2.6
JMC [47]	46.3	46.3	15.5%	**39.7%**	6,373	90,914	657	1,114	0.8
STAM16 [48]	46.0	50.0	14.6%	43.6%	6,895	91,117	473	1,422	0.2
MHT_DAM [11]	45.8	46.1	16.2%	43.2%	6,412	91,758	590	781	0.8
EDMT [49]	45.3	47.9	17.0%	**39.9%**	11,122	87,890	639	946	1.8
QuadMOT16 [50]	44.1	38.3	14.6%	44.9%	6,388	94,775	745	1,096	1.8
CDA_DDALv2 [51]	43.9	45.1	10.7%	44.4%	6,450	95,175	676	1,795	0.5
DP_NMS [52]	26.2	31.2	4.1%	67.5%	**3,689**	130,557	**365**	**638**	**5.9**


(a)

(b)

(c)

Fig. 14. Qualitative tracking results on MOT16-01 downloaded from MOT website (gray bounding boxes are changed to green for clarity). Three keyframes (frame 200, 240, 280) are shown in the figures. The public detections provided by MOT Challenge 2016 are shown in (a). Tracking results are presented in (b) and (c).

high-level detections and low-level superpixels, so there are more nodes in the track tree to help associate targets when detections are missing. Due to the high penalty between the detection node and dummy node, the correct branch has a lower score than the others when using too many dummy nodes and a wrong trajectory is generated. In contrast, HAF

TABLE IV
RESULTS ON MOT CHALLENGE 2017 TEST(5/1/2018)

Method	MOTA	IDF1	MT	ML	FP	FN	IDS	FM	Hz
eHAF17 (ours)	**51.8**	**54.7**	**23.4%**	37.9%	33,212	**236,772**	**1,843**	**2,739**	0.7
MHT_DAM [11]	50.7	47.2	20.8%	36.9%	22,875	252,889	2,314	2,865	0.9
EDMT [49]	50.0	51.3	21.6%	36.3%	32,279	247,297	2,264	3,260	0.6
PHD_GSDL17 [53]	48.0	49.6	17.1%	**35.6%**	23,199	265,954	3,998	8,886	6.7
IOU17 [54]	45.5	39.4	15.7%	40.5%	**19,993**	281,643	5,988	7,404	**1522.9**

tracker has the ability to keep targets associated by superpixels nodes instead of only using dummy nodes, giving branches more reasonable scores.

F. Qualitative Analysis

We show a set of qualitative tracking results of HAF tracker in Fig. 14. In Fig. 14-(c), our method successfully tracks the person in the purple bounding box (ID = 2) through frames from 200 to 280. It is because our proposed heterogeneous association graph is built by both detection nodes and superpixels nodes. The targets can be kept associated by superpixels nodes when detections are missed. In contrast, MHT_DAM [11] fails to track him because his detections are missed in Fig. 14-(a). Although dummy nodes are used, they can not keep the hypothesis when long-term detection failure happens. In addition, the person in the green bounding box (ID = 1) in Fig. 14-(c) is tracked earlier by HAF than by MHT_DAM where the detector does not locate her yet (too small for the detector).

VI. CONCLUSION

We propose a heterogeneous association graph fusion approach for target association in multiple object tracking. Our main innovation is to build a fused heterogeneous association graph that combines high-level detections and low-level image evidence. We build track trees by the fused association graph, and MHT is used for solving it. In HAF tracker, targets are indicated by both detections and superpixels, and thus targets are available to be associated even when a long-term detection failure happens. To gain a reasonable segmentation results, we propose a motion based algorithm that can extract foreground superpixels in moving scenes. Compared with other methods, HAF is able to locate more targets in the scene where the detector fails. We significantly reduce false negatives while controlling the increase of false positives at an acceptable level and show state-of-the-art results on the MOT Challenge 2017.

REFERENCES

[1] C. Bibby and I. Reid, "Real-time tracking of multiple occluding objects using level sets," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1307–1314.

[2] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1198–1211, Jul. 2008.

[3] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid, "Joint tracking and segmentation of multiple targets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5397–5406.

[4] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[5] S.-K. Weng, C.-M. Kuo, and S.-K. Tu, "Video object tracking using adaptive Kalman filter," *J. Vis. Commun. Image Represent.*, vol. 17, no. 6, pp. 1190–1208, Dec. 2006.

[6] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1515–1522.

[7] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. Eur. Conf. Comput. Vis.*, vol. 3021, 2004, pp. 28–39.

[8] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *Proc. 19th IEEE Conf. Decis. Control Including Symp. Adapt. Processes*, Dec. 1980, pp. 807–812.

[9] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. AC-24, no. 6, pp. 843–854, Dec. 1979.

[10] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 2, pp. 138–150, Feb. 1996.

[11] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4696–4704.

[12] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[13] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.

[14] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[15] A. A. Butt and R. T. Collins, "Multi-target tracking by Lagrangian relaxation to min-cost network flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1846–1853.

[16] N. McLaughlin, J. M. Del Rincon, and P. Miller, "Enhancing linear programming with motion modeling for multi-target tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2015, pp. 71–77.

[17] V. Chari, S. Lacoste-Julien, I. Laptev, and J. Sivic, "On pairwise costs for network flow multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5537–5545.

[18] A. Milan, K. Schindler, and S. Roth, "Detection- and trajectory-level exclusion in multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3682–3689.

[19] D. Mitzel, E. Horbert, A. Ess, and B. Leibe, "Multi-person tracking with sparse detection and continuous segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 397–410.

[20] A. E. Elgammal and L. S. Davis, "Probabilistic framework for segmenting people under occlusion," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 145–152.

[21] T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1208–1221, Sep. 2004.

[22] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[23] N. T. Siebel and S. Maybank, "Fusion of multiple tracking algorithms for robust people tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 373–387.

[24] L. Wen, Z. Lei, M.-C. Chang, H. Qi, and S. Lyu, "Multi-camera multi-target tracking with space-time-view hyper-graph," *Int. J. Comput. Vis.*, vol. 122, no. 2, pp. 313–333, 2017.

[25] C. Zhang, Y. Huang, Z. Wang, H. Jiang, D. Yan, and J. Cheng, "Cross-camera multi-person tracking by leveraging fast graph mining algorithm," *Multimedia Tools Appl.*, no. 9, pp. 1–18, 2018.

[26] W. Liu, O. Camps, and M. Sznaier. (2017). "Multi-camera multi-object tracking." [Online]. Available: https://arxiv.org/abs/1709.07065

[27] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[28] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[29] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[30] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[31] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[32] M. Chen, X. Wei, Q. Yang, Q. Li, G. Wang, and M.-H. Yang, "Spatiotemporal GMM for background subtraction with superpixel hierarchy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1518–1525, Jun. 2017.

[33] X. Liu *et al.*, "Background subtraction using spatio-temporal group sparsity recovery," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1737–1751, Aug. 2017.

[34] J. Chang, D. Wei, and J. W. Fisher, III, "A video representation using temporal superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2051–2058.

[35] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[36] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[37] P. R. J. Östergård, "A new algorithm for the maximum-weight clique problem," *Nordic J. Comput.*, vol. 8, no. 4, pp. 424–436, 2001.

[38] S. Busygin, "A new trust region technique for the maximum weight clique problem," *Discrete Appl. Math.*, vol. 154, no. 15, pp. 2080–2096, Oct. 2006.

[39] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1603.00831

[40] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, no. 1, p. 246309, 2008.

[41] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 17–35.

[42] M. Van Droogenbroeck and O. Barnich, "ViBe: A disruptive method for background subtraction," in *Background Modeling and Foreground Detection for Video Surveillance*, T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, Eds. London, U.K.: Chapman & Hall, Jul. 2014, ch. 7, pp. 7.1–7.23. [Online]. Available: http://hdl.handle.net/2268/157176, doi: 10.1201/b17223-10.

[43] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3701–3710.

[44] E. Levinkov *et al.*, "Joint graph decomposition & node labeling: Problem, algorithms, applications," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1904–1912.

[45] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 300–311.

[46] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3029–3037.

[47] S. Tang, B. Andres, M. Andriluka, and B. Schiele, "Multi-person tracking by multicut and deep matching," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 100–111.

[48] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 4836–4845.

[49] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong, "Enhancing detection model for multiple hypothesis tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 2143–2152.

[50] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3786–3795.

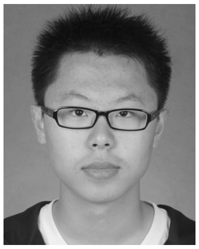
[51] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 595–610, Mar. 2018.

[52] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1201–1208.

[53] Z. Fu, P. Feng, F. Angelini, J. Chambers, and S. M. Naqvi, "Particle PHD filter based multiple human tracking using online group-structured dictionary learning," *IEEE Access*, vol. 6, pp. 14764–14778, 2018.

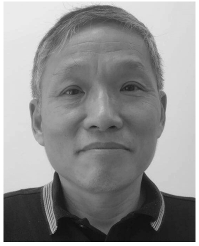
[54] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2017, pp. 1–6.

Hao Sheng received the B.S. and Ph.D. degrees from the School of Computer Science and Engineering, Beihang University, China, in 2003 and 2009, respectively, where he is currently an Associate Professor. His research interests include computer vision, pattern recognition, and machine learning.



Yang Zhang received the B.S. degree from the School of Advanced Engineering, Beihang University, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His research interest is computer vision, and he is particularly interested in multiple object tracking.

Jiahui Chen received the B.S. degree from the School of Advanced Engineering, Beihang University, China, in 2012, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His research interest is computer vision, and he is particularly interested in multiple object tracking.



Zhang Xiong received the B.S. degree from Harbin Engineering University in 1982 and the M.S. degree from Beihang University, Beijing, China, in 1985. He is currently a Professor with the School of Computer Science and Engineering, Beihang University. His research interests include computer vision, information security, and data vitalization.

Jun Zhang received the B.S. degree from the Harbin Shipbuilding Engineering Institute, Harbin, China, in 1982, and the M.S. and Ph.D. degrees from the Rensselaer Polytechnic Institute in 1985 and 1988, respectively. He was admitted to the graduate program of the Radio Electronic Department, Tsinghua University. He joined the Faculty of Department of Electrical Engineering and Computer Science, University of Wisconsin–Milwaukee, where he is currently a Professor. His research interests include image processing and signal processing.