This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

Online Scheme for Multiple Camera Multiple Target Tracking Based on Multiple Hypothesis Tracking

Haanju Yoo, *Member, IEEE,* Kikyung Kim, *Member, IEEE,* Moonsub Byeon, *Member, IEEE,* Younghan Jeon, *Member, IEEE,* and Jin Young Choi, *Member, IEEE*

***Abstract*—We propose an online tracking algorithm for multiple target tracking with multiple cameras. In this work, we suggest a multiple hypothesis tracking framework to find an unknown number of multiple tracks through the spatio-temporal association between tracklets generated from multiple cameras. In this framework, the multiple hypothesis tracking is realized online by solving the maximum weighted clique problem at every frame to estimate the three-dimensional trajectories of the targets. To handle the NP-hard issue of the maximum weighted clique problem, we propose a novel online scheme that formulates the maximum weighted clique problem using feedback information from the previous frame's result to find optimal tracks at every frame. This scheme enables the maximum weighted clique problem to be formulated by multiple subproblems and will significantly reduce the computation. The experiments show that the proposed algorithm performs comparably with the state-of-the-art batch algorithms, even though it adopts an online scheme.**

***Index Terms*—Multiple camera tracking, multiple hypothesis tracking, data association**

I. Introduction

MULTIPLE target tracking has been studied intensively as an essential technique in computer vision [1]–[10]. A number of online and batch methods that utilize the target's appearance and dynamic information have been proposed recently. However, these methods suffer from occlusion during visual surveillance because they use only a single camera. In particular, their performance drops in crowded settings, such as classrooms or stores, where targets are distributed densely and occluded by various obstacles. To overcome the occlusion problem, Possegger et al. [8] proposed a method that uses prior information about obstacles in the scene to reason about occlusions. Despite its robust performance with fixed obstacles, their method cannot be guaranteed to perform well with a scene containing moving obstacles.

Haanju Yoo, Kikyung Kim, Moonsub Byeon and Younghan Jeon are with Automation and Systems Research Institute (ASRI), Seoul National University, Seoul, 08826 Republic of Korea (South Korea) (e-mail: neohanju@snu.ac.kr; koreaton@snu.ac.kr; msbyeon@snu.ac.kr; yh1992@snu.ac.kr).

Jin Young Choi is with the Department of Electrical and Computer Engineering, Seoul National University, Seoul, 08826 Republic of Korea (South Korea) e-mail: (jychoi@snu.ac.kr).

To resolve the occlusion problem in multiple target tracking, methods using multiple cameras with overlapping fields of views have been proposed [11]–[21] (see Section II, Related Works). In the multiple camera multiple target tracking (MCMTT) problem, we have to jointly solve the problems of spatial and temporal association. In spatial association, the different viewed and concurrent observations from the same target are associated (i.e., reconstruction), while in temporal association, each camera's observations from the same target are associated through frames (i.e., tracking). Thus, MCMTT is more complex and difficult than single camera tracking. For this reason (as described in Section II, Related Works), most of the recent MCMTT algorithms are batch-based and require many computations. Despite their good performance, the existing MCMTT algorithms cannot be applied to actual systems, which need online and real-time processing.

In this paper, we propose an online algorithm for the MCMTT problem. In general, the online scheme has a crucial limitation that has fewer chances to recover the missing detections than the batch scheme. The missing detections occur frequently in a crowded pedestrian scene. To resolve a tracking ambiguity problem arising from densely distributed targets, we adopt the multiple hypothesis tracking (MHT) framework [22] based on a deferred decision. In MHT, track hypotheses (or tracks), which are the estimated trajectories of targets, are enumerated with all possible data associations between input measurements. Thus, the computational complexity of MHT grows when the number of input measurements increases. Although Kim et al. [23] showed that MHT performs well in single camera multiple target tracking, it is still challenging to moderate the computational complexity of MHT with measurements from multiple cameras.

To reduce the computational load of MHT with multiple cameras, we use tracklets—partial fragments of estimated target trajectories—on the two-dimensional (2D) image coordinates. The tracklets are generated through temporal associations between detections from the consecutive frames. We propose a scheme to generate candidate tracks by associating tracklets with their motion and appearance information. In the proposed scheme, the tracklets in each view are assembled in three-dimensional (3D) space by a back projection based on a ground plane assumption. In a ground plane assumption, all targets are assumed

Copyright (c) 2016 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

to move on a 3D virtual plane called a ground plane. With this assumption and the camera network calibration information, we can get the 3D location for each tracklet without any triangulation. Thus, our 3D association problem is simplified to a 2D association problem on the ground plane.

To find the set of tracks that best describes the tracking of targets among the candidate tracks, the association problem for MHT framework is formulated as the maximum weighted clique problem (MWCP), finding a complete subgraph of an arbitrary, undirected graph with the maximum total weights of its edges or vertices. To construct the graph in our MHT framework, we extend the graph from a single camera case in [24] and [25] to a multiple camera case. Our formulation introduces additional compatibility conditions to prevent ID switches between densely distributed targets and weights on vertices, which are assigned according to our carefully designed score function for candidate 3D tracks. Unlike the scores used in state-of-the-art MCMTT methods [16], [17], which consider only geometric information, our score takes into account not only geometric information but also motion and appearance information.

The MWCP is a well-known NP-hard problem, and as Kim et al. [23] showed, it is hard to find an exact solution during a limited time even when a graph is constructed with a single camera. In this paper, we propose a novel online scheme combining a heuristic MWCP algorithm with the divide-and-conquer algorithm, which is based on the feedback information from the K-best previous solutions. The proposed scheme moderates computational complexity as well as performance. Unlike the batch-based K-best approaches for a single camera case [15], [24] which find just a high scored track at each iteration and get the K-best tracks during K iterations, our online scheme is designed to find K sets of compatible tracks that best describe the multiple target tracking at each frame in multiple camera settings. This strategy helps our scheme to find a near optimum solution by the proposed online scheme with the divide-and-conquer algorithm based on feedback information. The feedback information from the results of the past frame makes a major contribution to the construction of tracklets, generate candidate tracks, and divide/conquer MWCP formulation. To construct tracklets, association conditions are designed to link the current detections and past tracklets. To generate candidate tracks, a track tree is proposed to link the current tracklets to the past candidate tracks. In addition, to reduce computation, the MWCP is reformulated into multiple subproblems based on the tracking solutions from the previous frame.

Even if we divide the original problem into subproblems, it is still challenging to solve each problem with an exact algorithm within a reasonable time for practical applications. For further computation reduction in solving each subproblem, we apply an iterative heuristic algorithm called breakout local search (BSL) [26], which is a state-of-the-art heuristic algorithm for MWCP. BLS not only finds a near-optimal solution rapidly but also generates multiple local optimum solutions for our online scheme when it is slightly modified. After solving these subproblems, the resultant solutions are also utilized in our pruning scheme to remove unreliable tracks.

II. Related Works

The MCMTT algorithms are categorized into three groups: reconstruction-and-tracking methods, tracking-and-reconstruction methods, and unified frameworks.

A. Reconstruction-and-tracking methods

The algorithms in this category have aimed to overcome the missing problem of object detection, caused by occlusion or background clutter, with measurements from additional views. At each frame, they integrate measurements from each view to generate probability maps of the existence of targets. They then apply a single camera-based tracking method on these maps to generate trajectories of targets. Fleuret et al. [11] and Khan et al. [13] proposed the probabilistic occupancy map (POM) and the synergy map, respectively. Those are the probability maps that estimate 3D locations of targets with a ground plane assumption. Many recent algorithms [14], [15] adopted them because of their robust performance in moderated scenarios. However, the POM has a limitation caused by the quantization of a tracking area and the synergy map has a ghost (or phantom) problem, which causes the ambiguity of a target's location. Possegger et al. [20] proposed a volumetric density map, which generalizes the ground plane assumption and moderates the above two problems. However, including [20], the three methods mentioned are very sensitive to the result of background subtraction, which is used in the generation of the probability maps. Hence, their algorithms suffer from scenes that have dynamic or complex backgrounds.

B. Tracking-and-reconstruction methods

In this category, the existing algorithms have attempted to associate trajectories of the same target, but from different views. They first independently generate trajectories at each view with a single camera-based tracking method, and then formulate a combinatorial problem to associate those trajectories. Wu et al. [18] formulated the trajectory association problem as a multidimensional assignment problem, which is a well-known NP-hard problem. They solved the problem with a heuristic algorithm named a greedy randomized adaptive local search procedure [27]. However, the method in [18] can only handle short-term occlusion because it is originally proposed to track hundreds of flying objects observed as point measurements, which occlude each other in a moment. Ayazoglu et al. [19] adopted a high order dynamic model in across view association of trajectories. The algorithm has a robust performance without calibration information even though targets have similar appearances. However, the algorithm suffers from a lot of computations because a comparison

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

between high-order dynamic models needs an operation that finds the rank of a huge matrix.

C. Unified frameworks

Many of the recent studies have proposed a unified framework, which formulates the reconstruction and the tracking problem into one unified global optimization problem, to solve those two problems jointly. The framework achieved good performance in various scenarios with a batch processing over a whole input video sequence. Leal-Taixé et al. [16] tried MCMTT algorithm with a unified framework for the first time. The framework constructs a 2D tracking graph of each camera with detections from the camera, and constructs a 3D reconstruction graph with pairs of detections from different cameras. Then, an optimization problem is solved over two graphs as one unified min-cost flow formulation. However, the proposed graph structure is too complicated and a rough estimate of the number of targets is needs as prior information to construct the graph. To resolve those problems, Hofmann et al. [17] proposed a method based on a hypergraph which can represent the reconstruction and tracking problem with a single graph. In this approach, all possible reconstructions between simultaneous detections must be enumerated to construct the graph. Furthermore, the approach solves the graph by a binary integer problem formulation, a well-known NP-hard problem, and its exact solver. Thus, the algorithm has a severe computational load.

Despite the robust performance on benchmarks, including crowds of more than ten people, the algorithms in [16], [17], [21] are all batch-based algorithms, so they cannot provide an instant tracking result at each frame, and require a huge number of computations. It is a serious limitation for many applications. Moreover, the algorithms use only the geometric information of measurements. Thus, they have difficulties in the consistent labeling when the proximity of targets increases. In this paper, we aim to propose an efficient online MCMTT algorithm without loss of performance, which is comparable to those state-of-the-art batch algorithms.

III. Problem Statements

The goal of our algorithm is to estimate trajectories of multiple targets from the given object detections in an online manner. A set of detections from all cameras is denoted by $\mathbf{D} = \{d_i | d_i = (l_i, s_i, c_i, t_i), i = 1, ..., N_{\mathbf{D}}\}$ where l_i, s_i indicate image coordinate location and scale, respectively, $c_i \in \{1, ..., N_C\}$ is a camera index, and t_i represents the time stamp when d_i is detected. $N_{\mathbf{D}}$ is the total number of input detections. A 2D tracklet is defined by a set of detections which are regarded as the successive measurements from the same target by the same camera. We define $\mathcal{Y}_j \in \mathbf{Y}$ as a j_{th} 2D tracklet by

$$\mathcal{Y}_j = \{d_i | i \in \mathbf{I}_{\mathcal{Y}_j}\}, \tag{1}$$

where $\mathbf{I}_{\mathcal{Y}_j}$ is an index set of the detections which belong to $\mathcal{Y}_j$. We assume that each detection cannot be shared by more than one target, which means that $\mathbf{I}_{\mathcal{Y}_i} \cap \mathbf{I}_{\mathcal{Y}_j} = \phi$ is always satisfied for $i \neq j$. Details on the generation of 2D tracklets from the given detections will be discussed in Section IV.

A track is an estimated trajectory of a target in 3D world coordinates. It is generated by associating tracklets presumed to be generated from the same target. When we define $\mathbf{I}_{\mathcal{T}_k}$ as an index set of the tracklets associated to a track $\mathcal{T}_k$, the detection set of $\mathcal{T}_k$, $\mathbf{Z}_k \subset \mathbf{D}$, can be defined with the 2D tracklets in $\mathbf{I}_{\mathcal{T}_k}$ as

$$\mathbf{Z}_k = \bigcup_{j \in \mathbf{I}_{\mathcal{T}_k}} \mathcal{Y}_j. \tag{2}$$

Let us define $\mathbf{Z}_k^t = \{d_i | d_i \in \mathbf{Z}_k, t_i = t\}$ as a set of the track's detections observed at time t from all cameras. Letting x_k^t be the estimated 3D location of target at time t, a track $\mathcal{T}_k \in \mathbf{T}$ is defined as the sequence of these estimated locations:

$$\mathcal{T}_k = (x_k^{t_k^s}, x_k^{t_k^s+1}, ..., x_k^{t_k^e}), \tag{3}$$

where $t_k^s = \min(\{t_i | d_i \in \mathbf{Z}_k\})$ and $t_k^e = \max(\{t_i | d_i \in \mathbf{Z}_k\})$ are the initiating and the terminating time of $\mathcal{T}_k$, respectively. In Section V, we will describe the details on the estimation of x_k^t from $\mathbf{Z}_k^t$ and the design of track $\mathcal{T}_k$'s score $S_{\mathcal{T}_k}$.

A global hypothesis $\mathcal{H}_n \in \mathbf{H}$ is a set of estimated trajectories of multiple targets, i.e., a subset of $\mathbf{T}$. When we define $\mathbf{I}_{\mathcal{H}_n}$ as an index set of the tracks belong to $\mathcal{H}_n$, then $\mathcal{H}_n$ is defined by

$$\mathcal{H}_n = \{\mathcal{T}_k | k \in \mathbf{I}_{\mathcal{H}_n}\}. \tag{4}$$

For feasible global hypotheses, any two different tracks $\mathcal{T}_k$, $\mathcal{T}_l$ belonging to the same global hypothesis must satisfy the compatibility conditions given by:

1) no common tracklet in any two tracks:
$$\mathbf{I}_{\mathcal{T}_k} \cap \mathbf{I}_{\mathcal{T}_l} = \phi, \tag{5}$$

2) collision avoidance:
$$|x_k^t - x_l^t| \geq \theta_s, \ \ \forall t \in [\max(t_k^s, t_l^s), \min(t_k^e, t_l^e)], \tag{6}$$

where θ_s means the minimum distance required in order to avoid a collision between targets. Here, we define the compatibility set $\mathbb{C}$ which consists of unordered index pairs of compatible tracks, that is, $\{k, l\} \in \mathbb{C}$ for all k, l satisfying the above conditions. Multiple target tracking is to find the best global hypothesis $\mathcal{H}_*$ which has the maximum total score among feasible global hypotheses satisfying the compatibility conditions:

$$\begin{aligned} \mathcal{H}_* = \arg\max_{\mathcal{H}_n} & \sum_{\mathcal{T}_k \in \mathcal{H}_n} S_{\mathcal{T}_k} \\ & s.t. \ \{k, l\} \in \mathbb{C}, \quad \forall k, l \in \mathbf{I}_{\mathcal{H}_n}. \end{aligned} \tag{7}$$

Since the problem in (7) is an NP-hard problem, in this paper, we aim to propose a novel online scheme to find a near-optimal solution of (7) at every frame by utilizing the past solutions. Fig. 1 depicts an overall scheme of the proposed method. It consists of four parts: tracklet, track, global hypothesis, and pruning.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

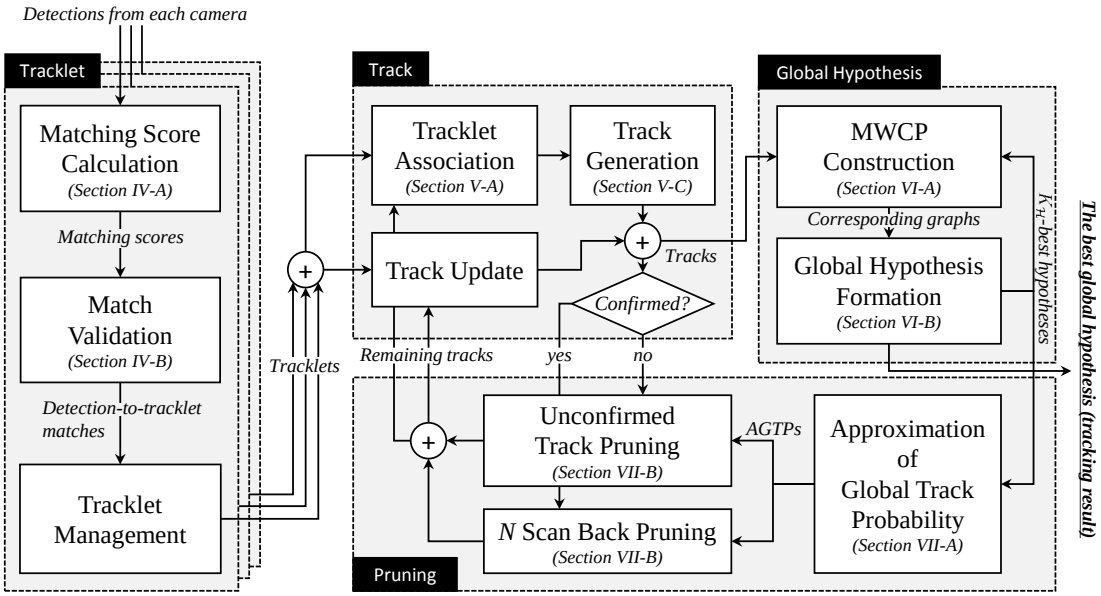

Fig. 1. Overall scheme of the proposed method. The arrows indicate the information flows. The proposed online scheme utilizes the past tracking results in each frame to find the current tracking result.

At each camera, the tracklet part generates tracklets by associating detections through frames. To associate detections in an online manner, we formulate a detection-to-tracklet matching problem. Then, we generate new tracklets or update established tracklets with the matching result. The generated tracklets are passed to the track part. Details on the matching and the tracklet management will be described in Section IV.

The track part generates new candidate tracks with associations between tracklets and manages established tracks in an online manner. To reduce the number of candidate tracks, we check the proposed conditions for the spatial and temporal association between tracklets. Then, a score of each track is computed with a carefully designed score function. Details on the track generation procedure and the score function will be described in Section V.

In the global hypothesis part, we solve (7) for $\mathbf{T}^t$, the set of all tracks in the current frame. To reduce the computation, we generate subproblems of (7) by referring $\mathbf{H}^{t-1} = \{\mathcal{H}_1^{t-1}, , ..., \mathcal{H}_{K_\mathcal{H}}^{t-1}\}$ which is the set containing the $K_\mathcal{H}$ best global hypotheses in the previous frame according to their total score. Then, we solve the subproblems instead of the original problem. Each subproblem is MWCP for $\mathbf{T}_n^t \subset \mathbf{T}^t$, a set of the tracks, which are candidates of the current best global hypothesis, with an assumption that the previous best global hypothesis was $\mathcal{H}_n^{t-1}$. To resolve the NP-hard issue in solving each MWCP, we adopt BLS with proposing a good initial solution and a proper iteration number. We also modify BLS to generate multiple near-optimal solutions. After gathering all global hypotheses found by solving subproblems, we pick the $K_\mathcal{H}$ best global hypotheses into $\mathbf{H}^t$. $\mathbf{H}^t$ is stored in the global hypothesis part for the next frame, and is conveyed to the pruning part. Details on the construction of MWCPs and for solving each of them will be described in Section VI.

TABLE I
NOTATIONS

Symbol	Description
d_i	i_{th} detection at image coordinates l_i of camera c_i at time t_i with a scale s_i
$\mathbf{D}, N_D = \|\mathbf{D}\|$	set of all detections
$\mathbf{D}^t$	set of all detections which are detected at time t
$\mathcal{Y}_j \in \mathbf{Y}$	j_{th} tracklet
$\mathcal{Y}_j^t$	detection of $\mathcal{Y}_j$ at time t, i.e., $\mathcal{Y}_j \cap \mathbf{D}^t$
$\mathbf{Y}^t$	set of all tracklets continuing until time t
$\mathbf{I}_{\mathcal{Y}_j}$	index set of detections in $\mathcal{Y}_j$
$\mathcal{T}_k \in \mathbf{T}$	k_{th} track hypothesis
$\mathbf{T}^t$	set of all existing tracks at time t
x_k^t	estimated 3D location of $\mathcal{T}_k$ at time t
t_k^s, t_k^e	initiating and terminating time of $\mathcal{T}_k$, respectively
$S_{\mathcal{T}_k}$	score of $\mathcal{T}_k$
$\mathbf{Z}_k \subset \mathbf{D}$	set of detections associated to $\mathcal{T}_k$
$\mathbf{Z}_k^t$	set of detections at time t in $\mathbf{Z}_k$, i.e., $\mathbf{Z}_k \cap \mathbf{D}^t$
$\mathbf{I}_{\mathcal{T}_k}$	index set of tracklets associated to $\mathcal{T}_k$
$\mathbb{C}$	compatibility set containing unordered index pairs of all compatible tracks in $\mathbf{T}$
$\mathbb{C}^t$	compatibility set of $\mathbf{T}^t$ instead of $\mathbf{T}$
$\mathcal{H}_n \in \mathbf{H}$	n_{th} global hypothesis
$\mathcal{H}_*$	the best global hypothesis
$K_\mathcal{H}$	the maximum number of global hypotheses for the subproblem generation and track pruning
$\mathbf{H}^t$	set of $K_\mathcal{H}$ best global hypotheses of time t
$\mathbf{I}_{\mathcal{H}_n}$	index set of tracks in $\mathcal{H}_n$

In the pruning part, two pruning techniques are applied to tracks, depending on whether a track is confirmed or not. A track is confirmed when its duration is longer than a certain length. We compute an approximated global track probability (AGTP) of each track with $\mathbf{H}^t$ and use it in each pruning technique as a criterion. Then, the pruning information is passed to the track part for reducing the number of tracks in the next frame. The definition of AGTP and details on pruning techniques will be described

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

in Section VII.

Before proceeding to the following sections, we summarize our notable notations with the Table. I. In the table, we also present notations for an online scheme, which are essential for the rest of this paper. We separate the notations into four groups related with inputs, tracklets, tracks, and global hypotheses, respectively.

IV. Tracklet Generation

A tracklet was widely used as an intermediate solution or a mid-level input in many previous works [7], [28], [29]. In our framework, we also use tracklets to reduce the number of overall computations. However, in our framework, the robustness of tracklets is crucial because there is no strategy to recover from wrong tracklets in our association for tracking. In this section, we present how we generate tracklets robustly with associating detections through successive frames at each camera, as defined in Section III. Our tracklet generation is similar to the method proposed by Benfold at el. [28] that utilizes the KLT feature tracking algorithm [30] in forward and backward direction. However, to generate tracklets in an online manner, we reformulate the inter-frame association problem to a detection-to-tracklet matching problem with a newly defined matching score, which is presented in the following section. We also apply matching validations on matched detections and tracklets to enhance the robustness of tracklets.

A. Matching score

Let us define $\mathbf{D}^t = \{d_i | t_i = t\}$ as the detections newly detected at time t, and $\mathbf{Y}^{t-1}$ as the set of all tracklets of which the last detections are at time $t-1$, that is,

$$\mathbf{Y}^{t-1} = \{\mathcal{Y}_j | \max(\{t_i | i \in \mathbf{I}_{\mathcal{Y}_j}\}) = t-1\}. \quad (8)$$

Then, the matching score between a tracklet $\mathcal{Y}_j \in \mathbf{Y}^{t-1}$ and a detection $d_i \in \mathbf{D}^t$ is defined by

$$S_{j,i} = \frac{1}{L_c}\left(S_{box}(\hat{\mathcal{Y}}_j^t, d_i) + \sum_{n=1}^{L_c-1} S_{box}(\mathcal{Y}_j^{t-n}, \hat{d}_i^{t-n})\right), \quad (9)$$

$$S_{box}(d_p, d_q) = -\log\left(\frac{\|l_p - l_q\|_2}{(s_p + s_q)/2}\right), \quad (10)$$

where $\mathcal{Y}_j^t = \mathcal{Y}_j \cap \mathbf{D}^t$ is the detection at time t which is included by tracklet $\mathcal{Y}_j$. L_c is the length of the comparison interval. $\hat{d}_i^t$ and $\hat{\mathcal{Y}}_j^t$ are results of the bi-directional (forward and backward) tracking described in the following. $\hat{d}_i^{t'}$ is the estimation of d_i at time $t' = t_i \pm 1$ (+1: forward, −1: backward). We assume that the size of a target does not change abruptly between consecutive frames. Thus, $\hat{d}_i^{t'}$ is defined by

$$\hat{d}_i^{t'} = (l_i + \tilde{\delta}_i^{t'}, s_i, c_i, t'), \quad (11)$$

where $\tilde{\delta}_i^{t'}$ indicates the major disparity between t_i and t' that are estimated by the motion estimation described in the following. In the motion estimation, we extract feature points from d_i and track them with the KLT feature tracking algorithm [30] on the frame at time t'. Let us define $\delta_{i,j}^{t'}, j = 1, ..., q_i^{t'}$ as the disparity between the j_{th} successfully tracked feature point in d_i at time t_i and its tracking result at time t'. Then, the disparity set $\mathbf{\Delta}_i^{t'}$ is defined by

$$\mathbf{\Delta}_i^{t'} = \{\delta_{i,j}^{t'} | \left\|\delta_{i,j}^{t'}\right\|_2 > \delta_{min}\}, \quad (12)$$

where δ_{min} is a design parameter to reject the disparities from static feature points. When the cardinality of $\mathbf{\Delta}_i^{t'}$ is smaller than the half of the number of all tracked feature points, we determine that d_i does not move, so $\tilde{\delta}_i^{t'} = 0$. Otherwise, we find the major disparity which has the largest neighbor set defined by

$$\mathcal{N}_{i,j}^{t'} = \{\delta_{i,k}^{t'} | \forall \delta_{i,k}^{t'} \in \mathbf{\Delta}_i^{t'} \ s.t. \ ||\delta_{i,j}^{t'} - \delta_{i,k}^{t'}||_2 < w_\delta(s_i)\}, \quad (13)$$

where $w_\delta(s_i)$ indicates the neighbor window size which is proportioned to s_i. Then, the major disparity is

$$\tilde{\delta}_i^{t'} = \underset{\delta_{i,j}^{t'} \in \mathbf{\Delta}_i^{t'}}{\arg\max} |\mathcal{N}_{i,j}^{t'}|. \quad (14)$$

By using forward tracking in the above, $\hat{\mathcal{Y}}_j^t$ can obtained from $\mathcal{Y}_j^{t-1}$, i.e., the last detection of the tracklet in the previous frame.

B. Validation of matching

At each frame, we match detections and tracklets by the Hungarian method [31], with scores defined in (9). To enhance the robustness of tracklets, we validate each match with two 3D geometric conditions in the following. The first condition is about the distance in 3D space between the matched detection and the matched tracklet's last detection. We assume that the distance must be close enough when the match is valid. To measure the 3D distance between detections, we have to know the 3D position of each detected pedestrian with a single detection. We resolve the depth ambiguity arises from a single detection by assuming that all pedestrian move on a specific 3D plane as mentioned in Chapter I. With the assumption and a Tsai camera calibration model [32], we can define a back projection function $\Phi(d_i)$ transferring the image coordinates l_i of camera c_i to the coordinates on the 3D ground plane. If the match between the detection d_i and the tracklet $\mathcal{Y}_j$ is valid, d_i and $d_k = \mathcal{Y}_j^{t-1}$, the last detection of $\mathcal{Y}_j$, must satisfy the condition below

$$|\Phi(d_k) - \Phi(d_i)| \leq \varepsilon_\Phi, \quad (15)$$

where ε_Φ is an allowable maximum 3D distance between consecutive detections in the same tracklet and is a design parameter related with the frame rate of an input video and the average moving speed of targets.

The second condition is about the estimated height of a detected object. We assume that the height of target does not change abruptly, thus d_i and $d_k = \mathcal{Y}_j^{t-1}$ are likely to have similar heights when the match between them is valid. Let us define $\hbar$ as the function that estimates the height of a detected object. Then, d_i and d_k must satisfy

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

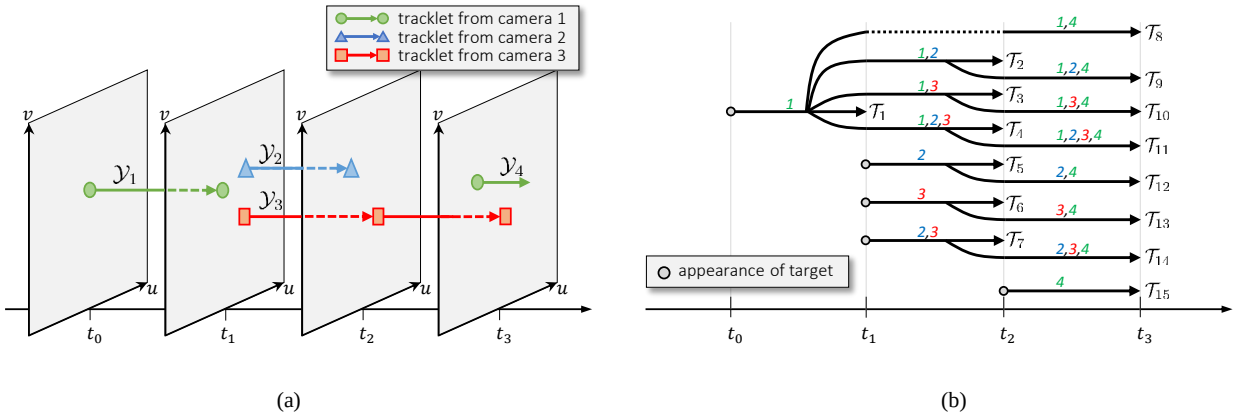

Fig. 2. Example showing the track generation. (a) (Back-projected) Tracklets from three cameras during four frames. The gray colored planes represent the ground plane in the world coordinates at each time. (b) Four track trees that are generated by associating tracklets in (a). The colored numbers on each track indicate the tracklets that are assigned to the track (i.e., *association set*). The dashed line in $\mathcal{T}_8$ represents the time gap in a temporal association, i.e., missing of detections.

the condition below to ensure the validity of the match between d_i and $\mathcal{Y}_j$

$$|\hbar(d_k) - \hbar(d_i)| \leq \varepsilon_{\hbar}, \tag{16}$$

where $\epsilon_{\hbar}$ is a design parameter about the maximum allowable variation in target's height between consecutive frames.

After the validation, we update tracklets with valid matches. If there is no matched detection for a tracklet, the tracklet is terminated. When there is no tracklet matched to a current detection, a new tracklet is generated with the detection.

V. Track Hypothesis

As mentioned in Section III, a track (hypothesis) is an estimated trajectory by combining tracklets from the same target. However, it is very challenging to determine ownerships of tracklets without any target information, including the exact number of targets. To resolve this, we generate all possible tracks through spatial-temporal association of tracklets until the current frame, and find the optimal tracks among them by solving the optimization problem in Section VI. In this section, we describe an online generation scheme of tracks and propose a track score representing the quality of each track. Furthermore, we also define a track tree which represents a hierarchical relationship between tracks.

A. Tracklet association

The data association between tracklets is to determine which tracklets are generated from the same target. Through the associations, the entire tracklets are partitioned into a multiple number of subsets. Each subset is assumed to be related to a target or a false alarm. A false alarm is the tracklet which is generated by non-target clutter. Let $\{\mathbf{\Omega}^1(\mathbf{Y}), \mathbf{\Omega}^2(\mathbf{Y}), ...\}$ be the collection of all possible partitions of an entire tracklet set $\mathbf{Y} = \{\mathcal{Y}_1, ..., \mathcal{Y}_q\}$. Then, the goal of our MCMTT problem is to find a partition $\mathbf{\Omega}^i(\mathbf{Y})$ which best describes the tracking of targets. The i_{th} partition is defined by $\mathbf{\Omega}^i(\mathbf{Y}) = \{\omega^i_{\phi}(\mathbf{Y}), \omega^i_1(\mathbf{Y}), ..., \omega^i_{n_i}(\mathbf{Y})\}$. $\omega^i_{\phi}(\mathbf{Y})$ is the set of false alarms which we call the *false alarm set*. $\omega^i_k(\mathbf{Y}), k = 1, ..., n_i$ is the set of tracklets supposed to be from the k_{th} target in the i_{th} partition, which we call the *association set*. Note that a track can be generated in a deterministic way when the corresponding association set is given. Thus, enumerating all possible association sets is equal to enumerating all possible tracks.

We define the universe set of all possible association sets without partition index as below

$$\begin{aligned} \mathbf{\Omega}(\mathbf{Y}) &= \bigcup_{i=1,2,...} \mathbf{\Omega}^i(\mathbf{Y}) \backslash \{\omega^i_{\phi}(\mathbf{Y})\} \\ &= \{\omega^1_1, ..., \omega^1_{n_1}, \omega^2_1 ...\} \\ &:= \{\omega_1, ..., \omega_{n_1}, \omega_{n_1+1}, \omega_{n_1+2}, ...\}. \end{aligned} \tag{17}$$

Here, we omit the argument '$(\mathbf{Y})$' for convenience. We notate the corresponding track of ω_k as $\mathcal{T}_k$ and the set of detections associated with ω_k as $\mathbf{Z}_k$. To describe our online generation scheme of association sets, we define three operations: *spatial association*, *temporal association* and *merge*.

1) Spatial Association: As $\mathcal{T}_2$ and $\mathcal{T}_7$ in Fig. 2(b), spatial association is defined by an association between temporally overlapping tracklets, which are supposed to be from the same target with different cameras. To determine whether tracklet $\mathcal{Y}_l$ and $\mathcal{Y}_m$ are from the same target or not, we propose the spatial association condition that checks if the distance between simultaneous detections in tracklets is bounded by ε_{3D} with a back projection function Φ described in Section IV-B, as below

$$\begin{aligned} &||\Phi(\mathcal{Y}^t_l) - \Phi(\mathcal{Y}^t_m)||_2 \leq \varepsilon_{3D}, \\ &\qquad \forall t \in \{t_i | \forall d_i \in \mathcal{Y}_l\} \cap \{t_j | \forall d_j \in \mathcal{Y}_m\}. \end{aligned} \tag{18}$$

2) Temporal Association: Temporal association is defined by an association between temporally not overlapped tracklets in the same camera or from different cameras that

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

are supposed to be from the same target. For temporal association, the preceding tracklet $\mathcal{Y}_l$ and the succeeding tracklet $\mathcal{Y}_m$ must satisfy following two conditions. *Condition 1*: According to the fact that each target can generate at most one detection in each view, $\mathcal{Y}_l$ and $\mathcal{Y}_m$ must be temporally non-overlapped. That is,

$$\{t_i | d_i \in \mathcal{Y}_l\} \cap \{t_j | d_j \in \mathcal{Y}_m\} = \phi. \tag{19}$$

Condition 2: A target cannot change its location abruptly, therefore d_p, the last detection of $\mathcal{Y}_l$, and d_n, the first detection of $\mathcal{Y}_m$, are close enough in 3D space. When v_{max} is defined as the maximum distance that a target can move during one frame, the condition is given by

$$\|\Phi(d_p) - \Phi(d_n)\|_2 \leq v_{max} \times |t_p - t_n|. \tag{20}$$

3) Merge: If the union set of two association sets satisfies all of spatial and temporal association conditions, the union set is also an association set. These two association sets are called as *mergeable*. Based on this, the merging operation '$\oplus$' between them is defined by

$$\omega_i \oplus \omega_j = \begin{cases} \omega_i \cup \omega_j, & \omega_i \text{ and } \omega_j \text{ are mergeable,} \\ \phi, & \text{otherwise.} \end{cases} \tag{21}$$

Since tracklets in each ω_i and ω_j already satisfy all association conditions, we only have to check the satisfaction of the conditions between a tracklet from ω_i and a tracklet from ω_j to determine whether the two association sets are mergeable or not.

B. Online generation of association sets

In this section, we describe how to generate $\mathbf{\Omega}^t$ with utilizing the set of association sets established until the previous frame $\mathbf{\Omega}^{t-1}$ and the set of newly generated tracklets at the current frame $\mathbf{Y}^t_{new} = \{\mathcal{Y}_i | \min\{t_j | d_j \in \mathcal{Y}_i\} = t\}$. At first, we generate $\mathbf{\Omega}^t_{new}$, the set of all possible association sets generated only with $\mathbf{Y}^t_{new}$. In $\mathbf{\Omega}^t_{new}$, there are association sets with a single tracklet in $\mathbf{Y}^t_{new}$ and association sets with at least two tracklets which satisfy the spatial association condition in (18). After generating $\mathbf{\Omega}^t_{new}$, we generate $\mathbf{\Omega}^t_{\oplus}$ by merging association sets in $\mathbf{\Omega}^{t-1}$ with mergeable association sets in $\mathbf{\Omega}^t_{new}$. But to reduce the computational complexity, we do not merge association sets if the frame gap between them is larger than δ_a. That is,

$$\begin{aligned} \mathbf{\Omega}^t_{\oplus} = \{\omega_i \oplus \omega_j | \forall \omega_i \in \mathbf{\Omega}^{t-1}, \forall \omega_j \in \mathbf{\Omega}^t_{new} \\ s.t.\ \omega_i \oplus \omega_j \neq \phi \text{ and } |t^s_j - t^e_i| \leq \delta_a\}. \end{aligned} \tag{22}$$

Then, $\mathbf{\Omega}^t$, the set of association sets established up to the current frame, is defined by

$$\mathbf{\Omega}^t = \mathbf{\Omega}^{t-1} \cup \mathbf{\Omega}^t_{new} \cup \mathbf{\Omega}^t_{\oplus}. \tag{23}$$

With $\mathbf{\Omega}^t$, we generate tracks by the method will be described in the next section.

If an association set $\omega_k \in \mathbf{\Omega}^t_{\oplus}$ is the result of a merge operation with $\omega_i \in \mathbf{\Omega}^{t-1}$ and $\omega_j \in \mathbf{\Omega}^t_{new}$, then ω_k and ω_i are two different association sets of the same target. Thus, their corresponding tracks $\mathcal{T}_k$ and $\mathcal{T}_i$ are incompatible and they cannot become optimal tracks at the same time. $\mathcal{T}_i$ is called the *parent track* of $\mathcal{T}_k$ whereas $\mathcal{T}_k$ becomes the *child track* of $\mathcal{T}_i$. A track is incompatible with not only its parent but also all of its ancestor tracks. Those incompatibilities between tracks are essential for the global hypotheses formation at Section VI and for the track pruning at Section VII. we depict those relationship with a hierarchical structure referred to as a *track tree*.

C. Track generation

When an association set ω_k is given, we can determine the detection set of $\mathcal{T}_k$ at time t, $\mathbf{Z}^t_k = \{d_i | t_i = t \text{ and } d_i \in \mathcal{Y}_j \text{ where } \mathcal{Y}_j \in \omega_k\}$. With $\mathbf{Z}^t_k$, the location of track $\mathcal{T}_k$ at time t, x^t_k, is estimated by two steps: *reconstruction* and *smoothing*. In this section, the definition of *reconstruction* is to generate the estimated 3D location $\hat{x}^t_k$ of a track $\mathcal{T}_k$ at time $t \in [t^s_k, t^e_k]$. When $\mathbf{Z}^t_k$ is a non-empty set, $\hat{x}^t_k$ is defined by the geometric center point of $\mathbf{Z}^t_k$ as below

$$\hat{x}^t_k = \frac{1}{|\mathbf{Z}^t_k|} \sum_{d_i \in \mathbf{Z}^t_k} \Phi(d_i). \tag{24}$$

If the target is not detected by any camera at time $t \in (t^s_k, t^e_k)$, $|\mathbf{Z}^t_k| = 0$. In this case, we estimate $\hat{x}^t_k$ by interpolation of the adjacent two reconstructed locations. Let us t_p and t_n denote the closest preceding and following time from t, which have a non-empty detection set, respectively. Then, the reconstructed 3D location at time t is defined by linear interpolation:

$$\hat{x}^t_k = \hat{x}^{t_p}_k + \frac{t - t_p}{|t_n - t_p|} \left(\hat{x}^{t_n}_k - \hat{x}^{t_p}_k \right). \tag{25}$$

$\hat{x}^t_k$ found in the reconstruction step is independent from other 3D locations of $\mathcal{T}_k$ at different times. However, adjacent locations are highly correlated with each other because the target moves under a specific motion. We do *smoothing* on reconstructed 3D locations to consider those dependencies. When all individual reconstructed 3D locations of $\mathcal{T}_k$ are found, x^t_k, the final 3D location of $\mathcal{T}_k$ at time t, is obtained by

$$x^t_k = \mathcal{F}((\hat{x}^{t^s_k}_k, ..., \hat{x}^{t^e_k}_k), t), \quad t = t^s_k, ..., t^e_k, \tag{26}$$

where '$\mathcal{F}(\cdot, t)$' is a function which returns the smoothed location at time t. We used SavitzkyâĂŞGolay filter [33] for this smoothing in our experiments.

D. Track score

We propose a score for each track while considering five factors. The first one is a reconstruction score $S_{\mathcal{R}}(\cdot)$ representing how the track's locations are identical to detections. The second one is a linking score $S_{\mathcal{L}}(\cdot)$ which considers the geometrical suitability of the consecutive locations of the track. The third and fourth ones are an initiation score $S_{\mathcal{I}}(\cdot)$ and a termination score $S_{\mathcal{T}}(\cdot)$. Each of them evaluates the suitability of the starting or the ending location of the track. When the track starts or ends

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

far from boundaries of the visible area or entrances, the track has a low initiation or termination score. The last one is a visual score $S_{\mathcal{V}}(\cdot)$ representing the visual similarity between detections associated to the track. Then, the track score is defined by those factors as

$$\begin{aligned} S_{\mathcal{T}_k} &= S\left(\mathcal{T}_k, \omega_k\right) \\ &= \sum_{t=t_k^s}^{t_k^e} S_{\mathcal{R}}\left(x_k^t, Z_k^t\right) + \sum_{t=t_k^s}^{t_k^e-1} S_{\mathcal{L}}\left(x_k^t, x_k^{t+1}\right) \\ &+ S_{\mathcal{I}}\left(x_k^{t_k^s}\right) + S_{\mathcal{T}}\left(x_k^{t_k^e}\right) + S_{\mathcal{V}}(\omega_k). \end{aligned} \tag{27}$$

1) Reconstruction Score $S_{\mathcal{R}}$: The proposed reconstruction score is based on $P_Z(\cdot)$, a likelihood of detection set on the target at a specific time. Letting X_k^t be the random variable standing for the location of the target tracked by $\mathcal{T}_k$ at time t, the reconstruction score is defined by

$$\begin{aligned} &S_{\mathcal{R}}\left(x_k^t, Z_k^t\right) \\ &\quad = \log\left(P_Z(Z_k^t | X_k^t = x_k^t)\right) - \log\left(P_Z(Z_k^t | X_k^t \neq x_k^t)\right). \end{aligned} \tag{28}$$

We give a penalty to the score with the second term when the target does not exist on Z_k^t at time t. This penalty term helps us to exclude false positive tracks in solving the MWCP in Section VI.

The likelihood is defined as:

$$P_Z(Z_k^t | X_k^t) = P_{vis}(Z_k^t | X_k^t) \times P_{rec}(Z_k^t | X_k^t), \tag{29}$$

where the first term is a visibility term representing the detection probability of Z_k^t, and the second term is a reconstruction term representing the error between Z_k^t and x_k^t in 3D space. Here, we assume that the two terms are independent of each other. The visibility term is defined in two circumstances depending on the existence of the target on x_k^t at time t. If the target exists on x_k^t, detections in Z_k^t are all true positives. By contrast, they are all false positives when the target does not exist on x_k^t. Letting γ_{fp} and γ_{fn} be the false positive ratio and the false negative ratio of the object detector, respectively, the visibility term of Z_k^t is defined by

$$P_{vis}(Z_k^t | X_k^t = x_k^t) = (1-\gamma_{fp})^{|Z_k^t|}(\gamma_{fn})^{n\left(x_k^t\right)-|Z_k^t|}, \tag{30}$$

$$P_{vis}(Z_k^t | X_k^t \neq x_k^t) = (\gamma_{fp})^{|Z_k^t|}(1-\gamma_{fn})^{n\left(x_k^t\right)-|Z_k^t|}, \tag{31}$$

where $n\left(x_k^t\right)$ indicates the number of cameras covering x_k^t by their field of views. When $|Z_k^t| > n\left(x_k^t\right)$, the track $\mathcal{T}_k$ is regarded as an invalid track. The reconstruction term $P_{rec}(Z_k^t | X_k^t)$ is based on the reconstruction error, which is defined by

$$\varepsilon_{rec}(x_k^t, Z_k^t) = \frac{1}{|Z_k^t|} \sum_{d_i \in Z_k^t} \left\|\Phi(d_i) - x_k^t\right\|_2. \tag{32}$$

To determine how large the allowable reconstruction error is, we borrow the maximum allowable reconstruction error from [5] which considers a calibration error and an object detection error as below

$$\varepsilon_{rec}^{max}(Z_k^t) = \varepsilon_{det} \cdot \sum_{d_i \in Z_k^t} \|\Theta(d_i)\| + \varepsilon_{cal}, \tag{33}$$

where ε_{det} represents the maximum allowable pixel error between the bottom center of the detection box and the actual grounding location of the target. ε_{cal} is a parameter about calibration error and it represents the maximum allowable distance between the back projections of each camera's image coordinates, which indicate the common 3D location, onto the 3D ground plane. The projection sensitivity function $\Theta(d_i)$ [5] indicates the variation of coordinates on the 3D ground plane, which is induced by one pixel variation around an image coordinates l_i of the camera c_i. Using (32) and (33), the reconstruction term of the likelihood is defined by

$$\begin{aligned} P_{\mathcal{R}_k^t} :=& P_{rec}(Z_k^t | X_k^t = x_k^t) \\ =& \begin{cases} \frac{1}{2}\, erfc\left(4\frac{\varepsilon_{rec}(x_k^t, Z_k^t)}{\varepsilon_{rec}^{max}(Z_k^t)} - 2\right), & |Z_k^t| > 1, \\ \frac{1}{2}, & \text{otherwise}, \end{cases} \end{aligned} \tag{34}$$

$$P_{rec}(Z_k^t | X_k^t \neq x_k^t) = 1 - P_{\mathcal{R}_k^t}. \tag{35}$$

where '$erfc(\cdot)$' is the complementary error function which is known as one minus the (Gauss) error function. '$erfc(\cdot)$' returns a large value on a small input, so the reconstruction term becomes larger when the reconstruction error of x_k^t is small. Here, we give 0.5 for the case of a single detection because it is impossible to get the reconstruction error, so we do not have any information about it. Using (28) and (30)-(34), the reconstruction score can be rewritten as

$$\begin{aligned} S_{\mathcal{R}}\left(x_k^t, Z_k^t\right) =& \log\left(\frac{P_Z(Z_k^t | X_k^t = x_k^t)}{P_Z(Z_k^t | X_k^t \neq x_k^t)}\right) \\ =& \left(n\left(Z_k^t\right) - |Z_k^t|\right) \times \log\left(\frac{\gamma_{fn}}{1-\gamma_{fn}}\right) \\ &+ |Z_k^t| \log\left(\frac{1-\gamma_{fp}}{\gamma_{fp}}\right) + \log\left(\frac{P_{\mathcal{R}_k^t}}{1-P_{\mathcal{R}_k^t}}\right). \end{aligned} \tag{36}$$

2) Linking Score $S_{\mathcal{L}}$: The motion of a pedestrian is hard to predict because it is usually non-linear. Therefore, we consider only the proximity of consecutive locations in the track to determine whether linking those locations is proper or not. The proposed linking score is defined to become bigger as the distance between consecutive locations in the track becomes smaller. That is,

$$S_{\mathcal{L}}\left(x_k^t, x_k^{t+1}\right) = \log\left(\frac{1}{2} erfc\left(4\frac{\left\|x_k^t - x_k^{t+1}\right\|_2}{v_{max}} - 2\right)\right), \tag{37}$$

where v_{max} is a design parameter modeling the maximum distance that a pedestrian can move during one frame. When $\left\|x_k^t - x_k^{t+1}\right\|_2$ is bigger than v_{max}, the linking is regarded as an invalid linking, so we discard the track.

3) Initiation/Termination Score $S_{\mathcal{I}}$, $S_{\mathcal{T}}$: As aforementioned, a track suddenly initiating or terminating in the middle of the visible area is less probable. To reflect this tendency, the initiation/termination scores are defined by

$$S_{\mathcal{I}}\left(x_k^t\right) = \log\left(P_s\right) - \tau_s \times \max\left(0, \mathcal{B}\left(x_k^t\right) - m_{\mathcal{B}}\right), \tag{38}$$

$$\begin{aligned} S_{\mathcal{T}}\left(x_k^t\right) =& \log\left(P_e\right) - \tau_e \times \max\left(0, \mathcal{B}\left(x_k^t\right) - m_{\mathcal{B}}\right) \\ &- \tau_l \times \left(t_k^e - t_k^s\right), \end{aligned} \tag{39}$$

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

where P_s and P_e are the probability of appearance and disappearance of a target in the visible area, respectively. τ_s and τ_e are coefficients of penalties with respect to the distance between the target and boundaries of the visible area. τ_l is a coefficient of a penalty that prevents the termination of long tracks. $\mathcal{B}(x_k^t)$ is the distance between x_k^t and boundaries of the visible area. $m_{\mathcal{B}}$ is the margin of boundary which is fixed to one meter in our experiments. A track which starts near the beginning frame is exempt from the initiation penalty, thus it has $\log(P_s)$ as its initiation score. With those initiation/termination scores, we can avoid the excessive number of fragmentations in the trajectories of targets.

4) Visual Similarity Score $S_{\mathcal{V}}$: We assume that a target does not change its appearance abruptly, so adjacent detections of the same track in each view should have similar appearances. To ensure this, we check the visual similarity between successive tracklets in the same view which are associated by temporal association. We assign the scores at temporal associations in ω_k of an arbitrary track $\mathcal{T}_k$. Letting $\omega_k^c = \{\mathcal{Y}_{k_1}^c, \mathcal{Y}_{k_2}^c, ..., \mathcal{Y}_{k_{n_{kc}}}^c\}$ be the ordered set of camera c's tracklets in ω_k according to their starting time, Ψ_k^c, the set of ordered pairs of detections is defined by

$$\begin{aligned}\Psi_k^c = \{(d_i, d_j) |\ d_i = \underset{d_q \in \mathcal{Y}_{k_l}^c}{\arg\max}\, t_q,\ d_j = \underset{d_p \in \mathcal{Y}_{k_{l+1}}^c}{\arg\min}\, t_p,\\ s.t.\ \mathcal{Y}_{k_l}^c, \mathcal{Y}_{k_{l+1}}^c \in \omega_k^c \text{ for } l = 1, ..., n_{kc} - 1\}.\end{aligned} \tag{40}$$

With Ψ_k^c and the function $\mathcal{V}(\cdot)$ extracting the visual feature from a detection, the visual similarity score is defined by

$$\begin{aligned}&S_{\mathcal{V}}(\omega_k) = \\ &-\sum_{c=1}^{C} \sum_{(d_i, d_j) \in \Psi_k^c} \alpha_v \times e^{\tau_v |t_j - t_i - 1|} \times \left\| \mathcal{V}(d_i) - \mathcal{V}(d_j) \right\|_2,\end{aligned} \tag{41}$$

where α_v and τ_v are parameters of modeling the declining confidence of visual similarity as the time gap between the detections increases. Since the goal of the score is to give a penalty to temporal associations between tracklets having inconsistent visual features, the visual similarity score is always less than or equal to zero.

VI. Global Hypothesis

A global hypothesis is a set of tracks generated by the partition of tracklets described in Section IV. However, as mentioned in Section III, a global hypothesis also can be defined as the set of compatible tracks. Thus, from a specific set of tracks, several global hypotheses can be generated by following compatibilities between the tracks. The goal of our tracking method is to find the best global hypothesis $\mathcal{H}_*$ among those global hypotheses, according to the track score defined in the previous section.

In this section, we present an online scheme which finds $\mathcal{H}_*$ rapidly. At every frame, we formulate MWCP as the optimization problem finding $\mathcal{H}_*^t$ among all possible global hypotheses from $\mathbf{T}^t$, the set of all tracks at time t. To reduce computation and enhance performance, we utilize $\mathcal{H}_*^{t-1}$, the optimal solution from the previous frame. However, it is easy to be trapped in a local optimum when we propagate only the best solution up to the current frame. To resolve this problem, we find not only the best solution but also the $K_{\mathcal{H}}$ best solutions $\mathbf{H}^t = \{\mathcal{H}_1^t, ..., \mathcal{H}_{K_{\mathcal{H}}}^t\}$ at each frame and use them to construct multiple MWCPs in the next frame. We also describe how we modify BLS, a state-of-the-art heuristic of solving MWCP, and apply it to our online scheme for the rapid generation of multiple solutions.

A. MWCP for MCMTT

At t_{th} frame, we construct $K_{\mathcal{H}}$ MWCPs that find global hypotheses which consist of high scored and compatible tracks. Each MWCP is constructed with $\mathbf{T}_n^t, n = 1, ..., K_{\mathcal{H}}$, a set of tracks which are candidates of the current best global hypothesis with an assumption that the previous best global hypothesis was $\mathcal{H}_n^{t-1} \in \mathbf{H}^{t-1}$. We call $\mathbf{T}_n^t$ a *related track set* of $\mathcal{H}_n^{t-1}$. It contains three types of tracks: (i) tracks in $\mathcal{H}_n^{t-1}$, (ii) tracks newly generated at the current frame among the children of (ii), i.e., $\mathbf{T}_{ch}^t(\mathcal{H}_n^{t-1})$, and (iii) unconfirmed tracks defined by

$$\mathbf{T}_{uc}^t = \{\mathcal{T}_i | \mathcal{T}_i \in \mathbf{T}^{\mathbf{t}} \text{ and } |t - t_i^s| < N_{conf}\}. \tag{42}$$

An unconfirmed track is a track shorter than N_{conf} frames and it is too short to determine whether it is a false positive or not. Thus, we constantly insert unconfirmed tracks into related track sets, even if those unconfirmed tracks are not in the previous solution. Then, $\mathbf{T}_n^t$ is given by

$$\mathbf{T}_n^t = \mathcal{H}_n^{t-1} \cup \mathbf{T}_{ch}^t(\mathcal{H}_n^{t-1}) \cup \mathbf{T}_{uc}^t. \tag{43}$$

MWCP for each $\mathbf{T}_n^t$ for $n = 1, ..., K_{\mathcal{H}}$ is formulated as

$$\begin{aligned}\boldsymbol{\tau}^* = \underset{\boldsymbol{\tau}}{\arg\max} \quad & \sum_{\mathcal{T}_i \in \mathbf{T}_n^t} S_{\mathcal{T}_i} \cdot \tau_i \\ s.t. \quad & \tau_i + \tau_j \leq 1, \quad \forall \{i, j\} \notin \mathbb{C}_n^t, \\ & \tau_i = 0, \quad \forall \mathcal{T}_i \notin \mathbf{T}_n^t, \\ & \tau_i \in \{0, 1\},\end{aligned} \tag{44}$$

where $\mathbb{C}_n^t = \{\{i, j\} | \mathcal{T}_i, \mathcal{T}_j \in \mathbf{T}_n^t, \{i, j\} \in \mathbb{C}, \}$ is the compatibility set of $\mathbf{T}_n^t$. The solution of the problem can be represented by the selection variable τ_i as $\mathcal{H}_{*,n}^t = \{\mathcal{T}_i | \tau_i^* = 1\}$. We solve each MWCP with the modified BLS described in Section VI-B, which quickly generates multiple locally optimal solutions from a single MWCP. Letting $\mathbf{H}_n^t$ be the locally optimal solutions found during solving MWCP for $\mathbf{T}_n^t$. Then, the entire feasible solutions found from all MWCPs in the current frame can be obtained by $\hat{\mathbf{H}}^t = \bigcup_{n=1,...,K_{\mathcal{H}}} \mathbf{H}_n^t$. We pick the $K_{\mathcal{H}}$ best solutions $\mathbf{H}^t$ from $\hat{\mathbf{H}}^t$ according to their total track scores. Then, the tracking solution of the current frame is the best one in $\mathbf{H}^t$. That is,

$$\mathcal{H}_*^t = \underset{\mathcal{H}_n^t \in \mathbf{H}^t}{\arg\max} \sum_{\mathcal{T}_k \in \mathcal{H}_n^t} S_{\mathcal{T}_k}. \tag{45}$$

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

When $\mathbf{H}^{t-1} = \phi$ as in the starting frame of an input video sequence, we construct and solve a single MWCP with an entire track set $\mathbf{T}^t$.

The computation of solving MWCP is exponentially proportional to the number of tracks when we solve it exactly. In ordinary cases, $|\mathbf{T}_n^t|$ is much smaller than $|\mathbf{T}^t|$. Thus, in general, our online scheme is faster than that solving the original problem even though the scheme has to solve multiple MWCPs.

B. BLS for MCMTT

BLS [26] is a state-of-the-art heuristic algorithm finding the maximum weighted clique in an undirected graph. BLS is based on an iteration which consists of a local search and a random perturbation. When it gets a locally optimal solution from the local search, it randomly perturbs the current solution to find another locally optimal solution. If the local search consecutively ends up with the same solution, BLS gradually increases its perturbation strength to escape from the local basin. This is called an adaptive perturbation, and is the key concept of BLS. In our online scheme, BLS is applied to solve each MWCP, with following three variants:

1) Multiple Solutions: Originally, BLS only keeps the best solution found at the moment. In contrast, we keep all the locally optimal solutions found until the algorithm meets the termination condition. Then, we pack those solutions into $\mathbf{H}_n^t$ and return it as the solving result of n_{th} MWCP.

2) Termination Condition: BLS uses only the maximum number of iterations as its termination condition because there is no way to guarantee the global optimality of a found solution. Thus, in [26], the maximum number of iterations was set to a huge constant to have more chance to find a better solution. However, it is intractable to the practical algorithms requiring real time requirements. We assume that the proper iteration number is in proportion to the complexity of the graph. And we assume that $|\mathbb{C}_n^t|$, a size of compatibility set, reflects the complexity of the graph. So, we propose the maximum iteration number i_{bls} as

$$i_{bls} = \alpha_{bls} \times |\mathbb{C}_n^t|, \tag{46}$$

where α_{bls} is a predetermined parameter and we set it to ten during all of our experiments. However, (46) has to be bounded for practical applications. Therefore, we saturate the maximum number of iterations with a predetermined parameter i_{bls}^{max}.

3) Initial Solution: In [26], BLS generates an initial solution $\mathcal{H}_{n_0}^t$ by random selection of compatible tracks. It is natural when there is no prior information. But in an online MCMTT, the solution from the previous frame can be a strong prior information to the current MWCP because targets move smoothly between consecutive frames. Therefore, we set $\mathcal{H}_{n_0}^t$ to $\mathcal{H}_n^{t-1}$ and perform a local search to refine an initial solution. However, the compatibilities between tracks can be changed as the tracking goes on, so we have to repair $\mathcal{H}_{n_0}^t$ before the local search when it is infeasible. The repairing of $\mathcal{H}_{n_0}^t$ is done in the following way: First, we set $\mathbf{T}_{cand}$, the candidate tracks for an initial solution, to $\mathcal{H}_n^{t-1}$. Then, insert a track with the highest track score in $\mathbf{T}_{cand}$ into $\mathcal{H}_{n_0}^t$ and update $\mathbf{T}_{cand}$ to $\{\mathcal{T}_i | \mathcal{T}_i \in \mathbf{T}_{cand} \backslash \mathcal{H}_{n_0}^t \ s.t. \ \forall \mathcal{T}_j \in \mathcal{H}_{n_0}^t, \{i,j\} \in \mathbb{C}_n^t\}$. Repeat those insertion and update until $\mathbf{T}_{cand}$ becomes an empty set.

VII. Pruning

In this section, we introduce our track pruning scheme to moderate the computational cost of our tracking algorithm. First, we compute each track's *approximated global track probability* (AGTP) which is proposed in the following subsection. After that, we apply two different pruning techniques depending on the confirmation of each track tree. For each unconfirmed track tree, the K-best method with AGTP is applied. Meanwhile, we adopt a classical pruning technique called N scan back approach [34] to pruning in each confirmed track tree.

A. Approximated global track probability

To score a track in a global view, [35] proposed a global track probability (GTP), defined with all global hypotheses that includes the track. When we define $\mathbb{H}^t$ as the set of all possible global hypotheses from $\mathbf{T}^t$, the GTP of track $\mathcal{T}_i \in \mathbf{T}^t$ is defined by

$$P_{\mathcal{T}}^t(\mathcal{T}_i) = \sum_{\forall \mathcal{H}_j \in \mathbb{H}^t s.t. \mathcal{H}_j \ni \mathcal{T}_i} P_{\mathcal{H}}^t(\mathcal{H}_j), \tag{47}$$

where $P_{\mathcal{H}}^t(\mathcal{H}_j)$ represents the probability of global hypothesis $\mathcal{H}_j \in \mathbb{H}^t$, which is defined by

$$P_{\mathcal{H}}^t(\mathcal{H}_j) = \frac{\sum_{\mathcal{T}_k \in \mathcal{H}_j} S_{\mathcal{T}_k}}{\sum_{\mathcal{H}_l \in \mathbb{H}^t} \sum_{\mathcal{T}_k \in \mathcal{H}_l} S_{\mathcal{T}_k}}. \tag{48}$$

Since finding $\mathbb{H}^t$ is intractable in most cases, calculating an exact GTP is intractable in general. Therefore, we approximate GTP with $\mathbf{H}^t$, the $K_{\mathcal{H}}$ best global hypotheses mentioned in Section VI, as below

$$\hat{P}_{\mathcal{T}}^t(\mathcal{T}_i) = \sum_{\forall \mathcal{H}_j \in \mathbf{H}^t s.t. \mathcal{H}_j \ni \mathcal{T}_i} \hat{P}_{\mathcal{H}}^t(\mathcal{H}_j), \tag{49}$$

$$\hat{P}_{\mathcal{H}}^t(\mathcal{H}_j) = \frac{\sum_{\mathcal{T}_k \in \mathcal{H}_j} S_{\mathcal{T}_k}}{\sum_{\mathcal{H}_l \in \mathbf{H}^t} \sum_{\mathcal{T}_k \in \mathcal{H}_l} S_{\mathcal{T}_k}}. \tag{50}$$

We prune tracks having zero AGTPs as the first step of our track pruning. Since AGTP is defined by the $K_{\mathcal{H}}$ best global hypotheses, a zero AGTP means that the track does not belong to any of the $K_{\mathcal{H}}$ best global hypotheses.

B. Track pruning scheme

Unconfirmed tracks in the same track tree are similar because their measurements are almost the same. Therefore, it is inefficient to keep all tracks in an unconfirmed track tree. Thus, we discard tracks all but the K_{uc} best tracks in each unconfirmed track tree according to their AGTPs. But to maintain the best global hypothesis, we also keep unconfirmed tracks in the best global hypothesis,

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

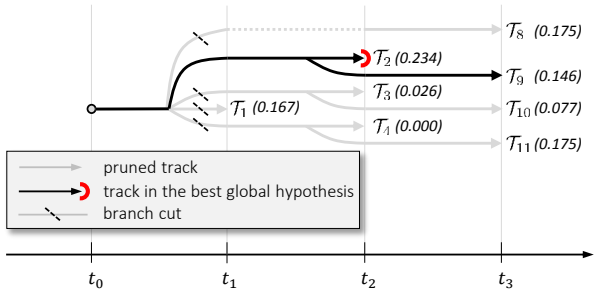

Fig. 3. Example of N scan back approach with confirmed track tree in Fig. 2(b) at t_3 (top one). In this example, we set $N = 3$. Parenthesized numbers are AGTPs. All branches are discarded except the branch containing a track in the best global hypothesis.

no matter that they are not included in the K_{uc} best tracks of their trees.

For the case of confirmed track trees, it is intractable to keep all tracks in each tree because the number of tracks in each tree increases exponentially. Thus, we apply N scan back approach [34] to confirmed track trees. It is the pruning technique based on a deferred decision. After finding the best global hypothesis, it scans N frames back and fixes locations of targets based on the current best global hypothesis. It then prunes all tracks incompatible with the fixed locations of targets. In view of a tree structure, it prunes branches of each track tree at N frames before, except the branch containing a track in the current best global hypothesis. Fig. 3 depicts an example of N scan back approach with one of the track trees in Fig. 2.

We summarize our online scheme by Algorithm 1.

Algorithm 1 Online scheme of MCMTT

Require: detections from the dataset $\mathbf{D}$
Ensure: tracking result $\mathbb{H}_* = \{\mathbf{H}_*^1, ..., \mathbf{H}_*^T\}$ ▷ T is the number of frames
$\mathbf{H}^0 \leftarrow \phi$
$\mathbf{Y}_c^0 \leftarrow \phi$ for $c = 1, ..., C$ ▷ C is the number of cameras
for t in $\{1, ..., T\}$ **do**
  for c in $\{1, ..., C\}$ **do**
    $\mathbf{D}_c^t \leftarrow \{d_i | c_i = c, t_i = t\}$
    $\mathbf{Y}_c^t \leftarrow generate\_tracklet(\mathbf{Y}_c^{t-1}, \mathbf{D}_c^t)$ ▷ sec. IV
  end for
  $\mathbf{Y}^t \leftarrow \bigcup_{c=1}^{C} \mathbf{Y}_c^t$
  $\mathbf{\Omega}^t \leftarrow tracklet\_association(\mathbf{\Omega}^{t-1}, \mathbf{Y}^t)$ ▷ sec. V-A
  $\mathbf{T}^t \leftarrow track\_generation(\mathbf{T}^{t-1}, \mathbf{\Omega}^t)$ ▷ sec. V-C
  if $\mathbf{H}^{t-1} = \phi$ **then**
    $\hat{\mathbf{H}}^t \leftarrow modified\_BLS(\mathbf{T}^t)$ ▷ sec. VI-B
  else
    for n in $\{1, ..., |\mathbf{H}^{t-1}|\}$ **do**
      $\mathbf{H}_n^t \leftarrow modified\_BLS(\mathbf{T}_n^t)$ ▷ sec. VI-B
    end for
    $\hat{\mathbf{H}}^t \leftarrow \bigcup_{n=1}^{|\mathbf{H}^{t-1}|} \mathbf{H}_n^t$
  end if
  $\mathbf{H}^t \leftarrow$ the $K_{\mathcal{H}}$ best global hypotheses in $\hat{\mathbf{H}}^t$
  $\mathbf{H}_*^t \leftarrow$ the best global hypothesis in $\mathbf{H}^t$
  $\mathbf{P}_{\mathcal{T}}^t \leftarrow calculate\_AGTP(\mathbf{T}^t, \mathbf{H}^t)$ ▷ sec. VII-A
  $\mathbf{T}^t \leftarrow track\_pruning(\mathbf{T}^t, \mathbf{P}_{\mathcal{T}}^t)$ ▷ sec. VII-B
  $\mathbf{\Omega}^t \leftarrow update\_association\_set(\mathbf{T}^t)$ ▷ collect association sets of remaining tracks
end for
$\mathbb{H}_* \leftarrow \bigcup_{t=1}^{T} \{\mathbf{H}_*^t\}$
return $\mathbb{H}_*$

VIII. Experiments

The proposed method was compared with the state-of-the-art MCMTT algorithms on the PETS 2009 dataset [36], the most widely used public benchmark dataset having multiple views from overlapping cameras. Also, the proposed method's design parameters were examined to see how they affect the trade-off between computation time and accuracy performance with a newly constructed PILSNU dataset[1]. Finally, the influence of the proposed track score's each term and the influence of the feedback information in the online scheme were analyzed with the PILSNU dataset.

1) Dataset: For the performance comparison, we used three sets in the second scenario of the PETS 2009 dataset: S2.L1, S2.L2, and S2.L3. These sets comprise four to seven different views from overlapping outdoor cameras. Each view has hundreds of frames capturing ten to 74 pedestrians at 7 fps. Each scenario has a different density of pedestrians, from low (S2.L1) to high (S2.L3). The benchmark dataset also provides intrinsic and extrinsic parameters of each camera with Tsai's camera model [32]. In our evaluation, we used the ground truth for the PETS 2009, provided in [37], which gives locations of targets in the region of interest in each frame.

[1]available on https://sites.google.com/site/neohanju/mcmtt

To examine the influence of parameters and terms in the proposed cost function on performance in accuracy, we constructed a new dataset, the PILSNU dataset. Our dataset contains 333 frames from each of four overlapping cameras capturing ten pedestrians whose are densely distributed in a small indoor environment at 6 fps. The dataset also provides Tsai's camera model for each camera, and a ground truth which is generated by hand labeled tracking result. For the fair comparison for following works, we also provide pedestrian detections obtained by the detector proposed by Nam at el. (LDCF detector) [38] for each frame of each camera. Since it is believed that Hofmann's algorithm [17] has the best performance among MCMTT batch algorithms, we also present the performance of Hofmann's on the PILSNU to give a reference for readers.

2) Evaluation metrics: As the metrics of a quantitative evaluation, we used multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) in the classification of events, activities, and relationships (CLEAR-MOT) metrics [39], the most popular metrics for MCMTT. We also used identity switches (IDS), fragments (FM), mostly tracked (MT), and mostly lost (ML), proposed by [40]. For those metrics, target locations in the ground truth are matched to the estimated locations in

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

TABLE II
PARAMETER SETTINGS

	Eq. no.	Symbol	Definition	Basis	Values (S2.L1, S2.L2, S2.L3, PILSNU)
Tracklet	(9)	L_c	length of the comparison interval	empirically	4 frames
	(12)	δ_{min}	threshold for static feature point	feature tracking error	0.1
	(15)	ε_Φ	maximum allowable 3D distance between consecutive detections	object moving speed	0.6 meters
	(16)	$\varepsilon_\hbar$	maximum allowable height variation of a target during one frame	pose changing speed	0.4 meters
	(13)	$w_\delta(s_i)$	neighbor window size for disparities	empirically	$0.2 \times s_i$
Track	(6)	θ_s	minimum 3D distance for collision avoidance	size of half body	0.2 meters
	(18)	ε_{3D}	maximum 3D distance between simultaneous detections	object moving speed & calibration error	2.0 meters
	(20), (37)	v_{max}	maximum 3D velocity of a target	normal speed of human	0.9 m/s
	(22)	δ_a	maximum allowable frame gap in a track	sufficient number	9 frames
	(30), (31)	γ_{fp}	false positive ratio of the object detector	from input detector	0.01
	(30), (31)	γ_{fn}	false negative ratio of the object detector	from input detector	(0.1, 0.4, 0.4, 0.2)
	(33)	ε_{det}	maximum error bound of detected location	from detection input	4 pixels
	(33)	ε_{cal}	maximum error bound of camera calibration	measured by reprojection	0.5 meters
	(38)	P_s	probability of target appearance	empirically	$(1.0^{-3}, 1.0^{-1}, 1.0^{-1}, 1.0^{-1})$
	(38)	τ_s	decaying coefficient of P_s w.r.t. a distance from boundary	empirically	$(1.0^{-3}, 1.0^{-4}, 1.0^{-4}, 1.0^{-3})$
	(39)	P_e	probability of target disappearance	empirically	$(1.0^{-6}, 1.0^{-1}, 1.0^{-1}, 1.0^{-2})$
	(39)	τ_e	decaying coefficient of P_e w.r.t. a distance from boundary	empirically	$(1.0^{-3}, 1.0^{-4}, 1.0^{-4}, 1.0^{-3})$
	(39)	τ_l	decaying coefficient of P_e w.r.t. a track length	empirically	$(1.0^{-2}, 1.0^{-2}, 1.0^{-2}, 1.0^{-1})$
Etc.	(42)	N_{conf}	minimum number of frames for track confirmation	empirically	3 frames
	-	i_{bls}^{max}	maximum iteration number of BLS	empirically	2,000
	-	K_{uc}	maximum number of tracks in each of unconfirmed track trees	empirically	4
	-	N	scan-back size	empirically	10 frames
	-	$K_\mathcal{H}$	number of global hypotheses	-	See Table III and Table IV

TABLE III
THE QUANTITATIVE RESULTS FOR THE PETS 2009 DATASET (SCENARIO 2)

Sequence	Method		Camera IDs	MOTA [%]	MOTP [%]	MT [%]	ML [%]	FM	IDS
PETS S2.L1	Batch	Berclaz et al. [14]	1+3+5+6+8	82.0	56.0	-	-	-	-
		Leal-Taixé et al. [16]	1+5	76.0	60.0	-	-	-	-
		Leal-Taixé et al. [16]	1+5+6	71.4	53.4	-	-	-	-
		Hofmann et al. [17]	1+5	99.4	82.9	**100.0**	**0.0**	1	1
		Hofmann et al. [17]	1+5+7	99.4	**83.0**	**100.0**	**0.0**	1	2
		Byeon et al. [21]	1+5+6+7+8	99.4	**83.0**	**100.0**	**0.0**	1	2
	Online	Ours (instant, $K_\mathcal{H} = 25$)	1+5+7	98.9	72.9	**100.0**	**0.0**	5	1
		Ours (deferred, $K_\mathcal{H} = 25$)	1+5+7	**99.5**	78.1	**100.0**	**0.0**	**0**	**0**
PETS S2.L2	Batch	Hofmann et al. [17]	1+2	**87.6**	73.5	**86.0**	**0.0**	**128**	**111**
		Hofmann et al. [17]	1+2+3	79.7	**74.2**	69.8	2.3	129	132
	Online	Ours (instant, $K_\mathcal{H} = 5$)	1+2	78.0	62.7	74.3	2.7	198	249
		Ours (deferred, $K_\mathcal{H} = 5$)	1+2	81.1	64.9	77.0	2.7	99	163
		Ours (instant, $K_\mathcal{H} = 10$)	1+2+3	69.5	61.0	75.7	2.7	220	357
		Ours (deferred, $K_\mathcal{H} = 10$)	1+2+3	72.9	63.1	73.0	2.7	132	246
PETS S2.L3	Batch	Hofmann et al. [17]	1+2	**68.5**	72.3	**54.5**	20.5	149	156
		Hofmann et al. [17]	1+2+4	65.4	**73.9**	40.9	25.0	88	116
	Online	Ours (instant, $K_\mathcal{H} = 20$)	1+2	63.6	59.1	43.2	20.5	87	119
		Ours (deferred, $K_\mathcal{H} = 20$)	1+2	64.4	59.9	40.9	18.2	**44**	**61**
		Ours (instant, $K_\mathcal{H} = 20$)	1+2+4	53.9	55.6	31.8	**9.1**	143	197
		Ours (deferred, $K_\mathcal{H} = 20$)	1+2+4	54.5	57.0	34.1	**9.1**	78	101

TABLE IV
THE QUANTITATIVE RESULTS FOR THE PILSNU DATASET

Sequence	Method		Camera IDs	MOTA [%]	MOTP [%]	MT [%]	ML [%]	FM	IDS
PILSNU	Batch	Hofmann et al.* [17]	1+2	61.3	77.6	60.0	**0.0**	8	12
		Hofmann et al.* [17]	1+2+3+4	**88.2**	**80.0**	80.0	**0.0**	**1**	**1**
	Online	Ours (instant, $K = 15$)	1+2	66.9	54.6	60.0	**0.0**	62	80
		Ours (deferred, $K_\mathcal{H} = 15$)	1+2	72.6	61.3	60.0	**0.0**	28	35
		Ours (instant, $K_\mathcal{H} = 10$)	1+2+3+4	80.0	64.4	**90.0**	**0.0**	32	44
		Ours (deferred, $K_\mathcal{H} = 10$)	1+2+3+4	85.7	72.5	**90.0**	**0.0**	12	18

the tracking result. A ground truth location is matched to the closest one among the estimated locations placed within one meter from the ground truth location.

3) Parameter settings: In our experiments, the degree and span size of the SavitzkyâĂŞGolay filter for smoothing in (26) were set to one and nine, respectively. The other parameters of the proposed method were set as shown in Table II. Although some parameters were empirically set, there are implications in using these parameters. When pedestrian density increases, the false negative ratio of a pedestrian detector goes up. In this case, a track's initiation or termination far from the boundary must be permitted because of occlusion. The initiation or termination controlled with P_s, P_e, τ_s and τ_e as shown in Table II.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

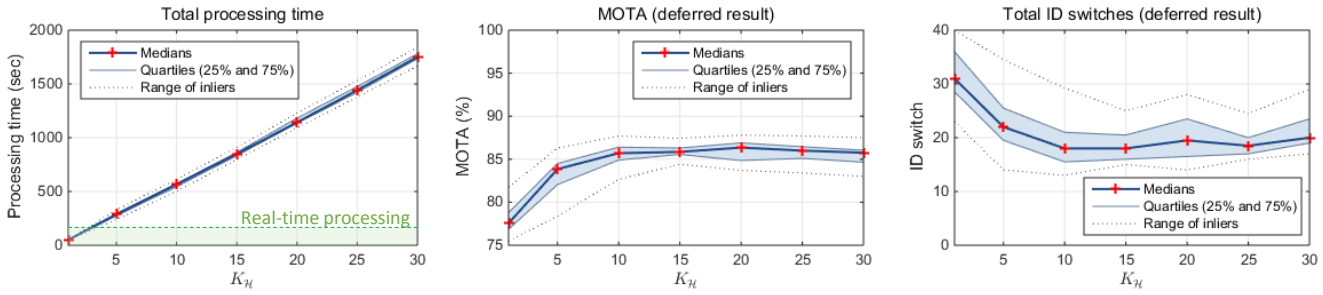

Fig. 4. Evaluation results with various values of $K_{\mathcal{H}}$ on the PILSNU dataset. The region between a lower quartile line (25%) and an upper quartile line (75%) is colored sky blue. Dashed lines represent the range of inlier defined by the maximum Whisker length, which is about ± 2.7 std. deviations from the mean value. In the left figure, we indicate a range of real-time processing by the green colored region.

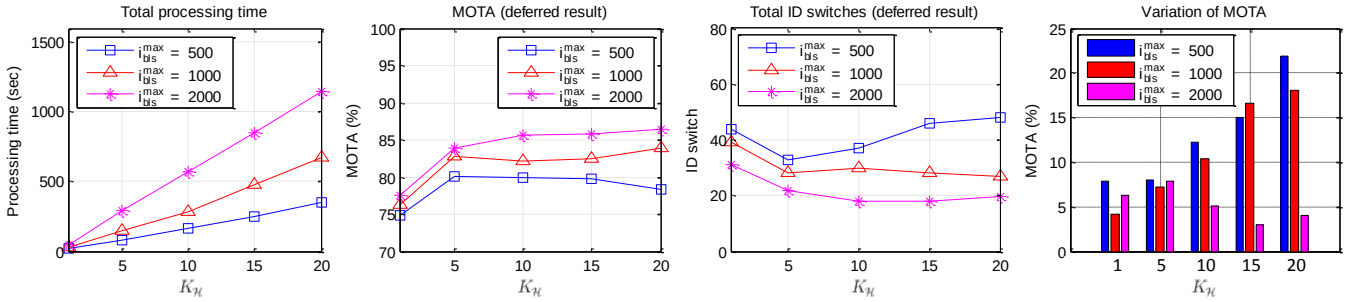

Fig. 5. Evaluation results with various values of i_{bls}^{max} and $K_{\mathcal{H}}$ on the PILSNU dataset. The last graph represents a gap between the maximum and minimum MOTA in each setting.

When a scene is sparse, the confidence of each track is higher than in a crowded scene. In this case, a smaller value of P_e than P_s is needed to prevent the impetuous termination of a well continuing track, which would be better than starting a new track.

4) Implementation details: We ran the experiments with a single threaded C++ implementation of our tracking algorithm on an i7 CPU, with 3.4 GHz, and 32 GB RAM. For the detection input of our algorithm, we used the deformable part model [41] for the PETS 2009, and LDCF for the PILSNU. As the visual feature in (41), we used a concatenated RGB histogram having 16 bins for each channel. In the tracklet generation, feature points were detected by FAST algorithm [42] combined with a grid adaptation technique to uniformly extract feature points from a target image.

A. Comparison with state-of-the-art methods

The proposed method was compared with the state-of-the-art MCMTT methods on the PEST 2009 dataset. Table III shows the results. The performance of existing works in the table have been cited from their papers. Since the performance may depend on the detection performance, we have conducted an additional experiment on our own PILSNU dataset including the detection results for fair comparison. In that experiment, our method was compared with Hofmann's algorithm [17], which has shown the best performance on the PETS 2009. For this experiment, we have implemented and tuned HofmannâĂŹs algorithm to perform comparably as it did in his paper. Table IV compares Hofmann's algorithm and ours on the PILSNU. The asterisks next to method names signify that we did the implementation ourselves and details would be different from the original implementation. Since our algorithm has randomness due to BLS, we ran 30 experiments for each sequence to get statistical results and present each metric with its median value. In regard to our results, "deferred" represents a ten frame deferred result, and "instant" represents an instant result without any delayed decision. In the parentheses, we present the value of $K_{\mathcal{H}}$ used in each sequence.

As shown in the Table IV, our deferred result is comparable to Hofmann's algorithm. Notably, our deferred result is superior to any other state-of-the-art batch methods in all metrics on the PETS 2009 S2.L1 except MOTP, which is significantly affected by a post processing. On S2.L1, our deferred result achieves 100% MT with zero IDS, which means a qualitatively perfect tracking result. Despite that our deferred result have much more IDS than Hofmann's algorithm on S2.L2, on which a pedestrian detector has low recall, the proposed framework achieves a comparable performance to the state-of-the-art batch algorithm although the online scheme has fewer chances to recover the missing detections than the batch scheme.

B. Influence of Parameters

On the PILSNU dataset, the proposed method was ran with $K_{\mathcal{H}} = 1$, 5, 10, 15, 20, 25, 30 and $i_{bls}^{max} = 500$, 1,000, 2,000 to examine the effect of $K_{\mathcal{H}}$ and i_{bls}^{max} on accuracy and computation time. Due to the randomness

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

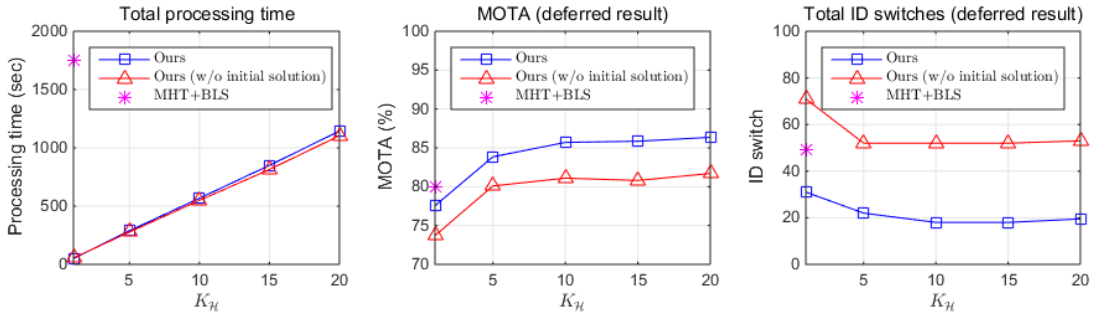

Fig. 6. Quantitative results with various solving schemes. "MHT+BLS" indicates the baseline scheme that solves (44), one MWCP with entire candidate tracks, without utilizing any feedback information on the previous global hypothesis. "Ours (w/o initial solution)" indicates the proposed scheme, but without the initial solution decribed in Section VI-B. Instead, it uses a random initial solution as the original BLS.

of the proposed method, 30 experiments were done at each parameter setting, as mentioned in the previous section. Fig. 4 shows the tendencies of the processing time, MOTA, and IDS affected by $K_{\mathcal{H}}$ in the proposed algorithm. The processing time increased linearly depending on $K_{\mathcal{H}}$. Thus, a desired processing time can be achieved by adjusting $K_{\mathcal{H}}$. MOTA increased with $K_{\mathcal{H}}$ and converged around 85%. IDS decreased until it touched 18 at $K_{\mathcal{H}} = 10$. Clearly, large values of $K_{\mathcal{H}}$ boost performance. Therefore, $K_{\mathcal{H}}$ is a conclusive control variable of the trade-off between performances in accuracy and computation times. Note that the PILSNU has 333 frames captured at 6 fps. Thus, processing the whole dataset within 55.5 seconds is the condition of real-time processing. Since our result at $K_{\mathcal{H}} = 1$ had a processing time of less than 50 seconds and shows a performance comparable with the state-of-the-art batch algorithms, we believe that the proposed algorithm has a real-time capability.

Fig. 5 shows the tendencies of the processing time, MOTA, variation at MOTA, and IDS affected by i_{bls}^{max}. i_{bls}^{max} is also crucial to the performance of our algorithm because (46) usually hits i_{bls}^{max}. As shown in the figure, the processing time also increased linearly depending on i_{bls}^{max}. However, the performance was not increased without a sufficiently large value of i_{bls}^{max}. The last graph in Fig. 5 represents the gap between the maximum and minimum MOTA in each setting. Following that graph, a small i_{bls}^{max} caused an unstable performance while MOTA with a sufficiently large i_{bls}^{max} was gradually stabilized according to $K_{\mathcal{H}}$.

C. Score Function Analysis

To examine the influence of each term in the proposed score function upon the performance of our algorithm, score function was evaluated with term variations: (i) without the visual similarity score, (41), (ii) without the visibility term in (29), (iii) without the reconstruction term in (29), (iv) with a constant score instead of (38) and (39), (v) with a constant score instead of (37). Table V provides the results and shows that the most crucial term was the linking score. With a constant linking score, only the visual similarity score measures the quality of a linkage between tracklets. However, the visual similarity score does not work with tracklets from different cameras; hence, the result is trivial. Except the linking score, the initiation and termination scores were the most crucial terms. These scores are deeply connected to the false positives and false negatives of target tracking. Thus, constant initiation and termination scores dropped down MT as constant linking scores did. The most uninfluential term was the visual similarity score; it only improved IDS and FM slightly. The influence of visibility on the performance was also negligible.

TABLE V
THE QUANTITATIVE RESULTS OF SCORE FUNCTION VARIATIONS

Description	MOTA	MOTP	MT	ML	FM	IDS
proposed	**85.7**	72.5	**90.0**	**0.0**	**12**	18
w/o visual similarity	85.4	72.2	**90.0**	**0.0**	15	19
w/o visibility term	84.9	**72.9**	**90.0**	**0.0**	15	**17**
w/o reconstruction term	81.4	71.2	**90.0**	**0.0**	19	29
constant init/term score	75.7	69.9	80.0	**0.0**	21	45
constant linking score	74.1	71.8	80.0	**0.0**	35	52

D. Solving Scheme Analysis

To verify the effectiveness of our proposed solving scheme, we compared our scheme to its variation and its baseline scheme. Fig. 6 shows the quantitative result of each solving schemes. "MHT+BLS" indicates the baseline solving scheme that uses only one MWCP for solving (44) without feedback information from the previous global hypotheses. Therefore, we plot the result of "MHT+BLS" with a single point at $K_{\mathcal{H}} = 1$. "Ours (w/o initial solution)" indicates the solving scheme that constructs multiple MWCPs with the feedback information, but uses a randomly generated initial solution as the original BLS. Except the baseline scheme, the maximum iteration number i_{bls}^{max} was set to 2,000. For a fair comparison, i_{bls}^{max} in the baseline scheme was set to $5 \times 2,000 = 10,000$. With $i_{bls}^{max} = 10{,}000$, the computation time of the baseline exceeded that of the other schemes.

As shown in Fig. 6, the baseline scheme was superior to the other schemes when $K_{\mathcal{H}} = 1$. This, however, is trivial because the other schemes solve smaller graphs than the baseline scheme, signifying that they cannot escape from the local optimum solution near or around the previous so-

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

lution. However, despite solving smaller graph rather than the baseline scheme, performance increased when $K_{\mathcal{H}}$ had a large value. The proposed scheme performed better than the baseline scheme when $K_{\mathcal{H}} = 5$ or greater. "Ours (w/o initial solution)" also achieved a comparable performance to the baseline scheme when $K_{\mathcal{H}} = 5$ or greater. Since the computation time of the proposed scheme is much smaller than the baseline scheme, it is certain that our problem dividing strategy is beneficial to performance when the computation time is limited. Compared to "Ours (w/o initial solution)", the performance of the proposed solving scheme improved MOTA by 5.5% and IDS by 61.6% on average. Thus, the effectiveness of the proposed initial solution is verified. Note that this result shows that the solutions for consecutive frames are closely related.

IX. Conclusion

We have presented an online MCMTT algorithm based on the MHT framework. Our MHT framework was realized by MWCP for the tracklet association to find multiple object tracks until the current frame. By using tracklets and proposed association conditions, our tracking algorithm generates refined candidate tracks and rejects a number of unreliable candidate tracks. To solve MWCP, we proposed an online scheme, which generates multiple MWCPs with small track sets collected according to the past global hypotheses in the previous frame. Those subproblems have much smaller solution search spaces than the original MWCP, so the proposed scheme significantly reduced the solving time. Moreover, the proposed scheme also found better solutions than solving MWCP with entire tracks. To resolve the NP-hard issue in solving each MWCP, we used BLS modified for MCMTT. The proposed initial solution and iteration number helped BLS to find near-optimal solutions with a small number of iterations. As shown in the experiments, our online MCMTT algorithm shows the state-of-the-art performance on the public benchmark dataset, and also shows the capability of real-time processing.

Acknowledgment

This work was supported by the IT R&D program of MOTIE/KEIT. [10041610, The development of automatic user information(identification, behavior, location) extraction and recognition technology based on perception sensor network(PSN) under real environment for intelligent robot]

References

[1] J. Yang, P. A. Vela, Z. Shi, and J. Teizer, "Probabilistic multiple people tracking through complex situations," in *11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[2] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 9, pp. 1820–1833, 2011.

[3] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1265–1272.

[4] Z. Wu, J. Zhang, and M. Betke, "Online motion agreement tracking," in *Proc. BMVC*, 2013.

[5] M. Hofmann, M. Haag, and G. Rigoll, "Unified hierarchical multi-object tracking using global data association," in *Performance Evaluation of Tracking and Surveillance (PETS), 2013 IEEE International Workshop on*. IEEE, 2013, pp. 22–28.

[6] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li, "Multiple target tracking based on undirected hierarchical relation hypergraph," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1282–1289.

[7] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1218–1225.

[8] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof, "Occlusion geodesics for online multi-object tracking," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1306–1313.

[9] A. Milan, K. Schindler, and S. Roth, "Detection-and trajectory-level exclusion in multiple object tracking," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3682–3689.

[10] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 1, pp. 58–72, 2014.

[11] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 267–282, 2008.

[12] Z. Wu, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke, "Tracking a large number of objects from multiple views," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1546–1553.

[13] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 3, pp. 505–519, 2009.

[14] J. Berclaz, F. Fleuret, and P. Fua, "Multiple object tracking using flow linear programming," in *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*. IEEE, 2009, pp. 1–8.

[15] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 9, pp. 1806–1819, 2011.

[16] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn, "Branch-and-price global optimization for multi-view multi-target tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1987–1994.

[17] M. Hofmann, D. Wolf, and G. Rigoll, "Hypergraphs for joint multi-view reconstruction and multi-object tracking," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3650–3657.

[18] Z. Wu, T. H. Kunz, and M. Betke, "Efficient track linking methods for track graphs using network-flow and set-cover techniques," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1185–1192.

[19] M. Ayazoglu, B. Li, C. Dicle, M. Sznaier, O. Camps *et al.*, "Dynamic subspace-based coordinated multicamera tracking," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2462–2469.

[20] H. Possegger, S. Sternig, T. Mauthner, P. M. Roth, and H. Bischof, "Robust real-time tracking of multiple objects by volumetric mass densities," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2395–2402.

[21] M. Byeon, S. Oh, K. Kim, H. J. Yoo, and J. Y. Choi, "Efficient spatio-temporal data association using multidimensional assignment for multi-camera multi-target tracking," in *Proc. BMVC*, 2015.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2016.2593619, IEEE Transactions on Circuits and Systems for Video Technology

[22] D. B. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843–854, 1979.

[23] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4696–4704.

[24] A. R. Zamir, A. Dehghan, and M. Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 343–356.

[25] D. J. Papageorgiou and M. R. Salpukas, "The maximum weight independent set problem for data association in multiple hypothesis tracking," in *Optimization and Cooperative Control Strategies*. Springer, 2009, pp. 235–255.

[26] U. Benlic and J.-K. Hao, "Breakout local search for maximum clique problems," *Computers & Operations Research*, vol. 40, no. 1, pp. 192–206, 2013.

[27] T. A. Feo, M. G. Resende, and S. H. Smith, "A greedy randomized adaptive search procedure for maximum independent set," *Operations Research*, vol. 42, no. 5, pp. 860–878, 1994.

[28] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3457–3464.

[29] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association with online target-specific metric learning," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1234–1241.

[30] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[31] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[32] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3d machine vision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1986*, 1986.

[33] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[34] Y. Bar-Shalom, "Multitarget-multisensor tracking: advanced applications," *Norwood, MA, Artech House, 1990, 391 p.*, vol. 1, 1990.

[35] S. S. Blackman, "Multiple-target tracking with radar applications," *Dedham, MA, Artech House, Inc., 1986, 463 p.*, vol. 1, 1986.

[36] A. Ellis, A. Shahrokni, and J. M. Ferryman, "Pets2009 and winter-pets 2009 results: A combined evaluation," in *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*. IEEE, 2009, pp. 1–8.

[37] A. Andriyenko, S. Roth, and K. Schindler, "An analytical formulation of global occlusion reasoning for multi-target tracking," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1839–1846.

[38] W. Nam, P. Dollár, and J. H. Han, "Local decorrelation for improved pedestrian detection," in *Advances in Neural Information Processing Systems*, 2014, pp. 424–432.

[39] 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," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 319–336, 2009.

[40] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2953–2960.

[41] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.

[42] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 430–443.

Haanju Yoo received the B.S. degree in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea, in 2009. Currently, he is a Ph.D. candidate in the Department of Electrical Engineering and Computer Science of Seoul National University, Seoul, Korea. His research interests include adaptive and learning systems, multi-camera multi-target tracking, object detection in crowded scenes, and so on.

Kikyung Kim received the B.S. and M.S. degrees in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea, in 2013 and 2015, respectively. Currently, he is a Ph.D. student in the Department of Electrical Engineering and Computer Science of Seoul National University, Seoul, Korea. His research interests include multi-camera multi-target tracking, object detection, image dehazing and so on.

Moonsub Byeon received the B.S. and M.S. degrees in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea, in 2010 and 2012, respectively. Currently, he is a Ph.D. candidate in the Department of Electrical Engineering and Computer Science of Seoul National University, Seoul, Korea. His research interests include multi-camera multi-target tracking, and so on.

Younghan Jeon received the B.S. degree in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea, in 2014. Currently, he is a Ph.D. student in the Department of Electrical and Computer Engineering of Seoul National University, Seoul, Korea. His research interests include multi-camera behavior understanding, and so on.

Jin Young Choi received the B.S., M.S. and Ph.D. degrees from Seoul National University, Seoul, Korea, in 1982, 1984, 1993, respectively. From 1984 to 1994, he has been at the Electronics and Telecommunication Research Institute (ETRI). Since 1994, he has been with Seoul National University, where he is currently Professor in the Department of Electrical and Computer Engineering. From 1998/8 to 1999/8, he was a Visiting Professor at University of California, Riverside. His research interests include adaptive and learning systems, object detection/tracking, and pattern recognition.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.