On-Road Pedestrian Tracking Across Multiple Driving Recorders

Kuan-Hui Lee and Jenq-Neng Hwang, *Fellow, IEEE*

***Abstract*—In this paper, we propose a new framework to track on-road pedestrians across multiple driving recorders. The framework is built upon the results of tracking under a single driving recorder. More specifically, we treat the problem as a multi-label classification task and determine whether a specific pedestrian belongs to one or several cameras' field of views by considering association likelihood of the tracked pedestrians. The likelihood is calculated based on the pedestrians' motion cues and appearance features, which are necessarily transformed via brightness transfer functions obtained by some available spatially overlapping views for compensating diversity of the cameras. When a pedestrian is leaving a camera's field of view, the proposed framework predicts and interpolates its possible moving trajectories, facilitated by open map service which can provide routing information. Experimental results show the robustness and effectiveness of the proposed framework in tracking pedestrians across several recorded driving videos. Moreover, based on the GPS locations, we can also reconstruct a 3-D visualization on a 3-D virtual real-world environment, so as to show the dynamic scenes of the recorded videos.**

***Index Terms*—3-D visualization, multi-label classification, pedestrian tracking, visual surveillance.**

I. Introduction

CURRENTLY, most of the surveillance cameras are installed at fixed locations, which reduce the flexibility of camera views and may result in blind-spots of monitoring. Thus, the idea of developing mobile surveillance systems is introduced. Fortunately, an emerging application of video analyses in autonomous/smart vehicles is the usage of the driving recorder (or dash-cameras), which is a device that records video in a vehicle to create a record of driving [1]. In addition to recording videos, a vehicle can also obtain other driving information such as location, time, and speed, by global positioning system (GPS) or other electronic sensing devices. Inspired by the growing usage of the driving recorders and advanced wireless infrastructure, we can soon expect a mobile surveillance platform with a cloud server being connected to all devices via wireless wide area network (WWAN), such as 3G, WiMAX, or LTE. Based on the collected data, the cloud server can thus systematically perform video analyses to better monitor the on-road situation dynamically and cooperatively share the analyzed on-road information. Such a mobile camera platform can be combined with the traditional surveillance systems, to establish a larger and wider intelligent surveillance system - a platform of collecting and analyzing large amounts of real-time correlated videos, i.e., *big visual data*. Based on the coordinated analyses of these big visual data from many cloud-connected static/mobile cameras, a 3-D visualization of dynamic on-road scenes can be further reconstructed as shown in Fig. 1. The purpose is not only to visualize the pedestrians' trajectories and movements in the 3-D environment but also to avoid the issues of privacy invasion of the pedestrians by using avatar-like 3-D models. When there is an event, such as accident or activity, we can see the dynamic scene from different aspects of views for further analyses.

Fig. 1. 3-D visualization of the scene recorded by four driving recorders. Each row belongs to one driving recorder; the leftmost is the video frames, the middle is the corresponding view of 3-D visualization on Google Earth, and the right is the scene visualized from different aspect.

Pedestrian tracking is one of the major topics in the video surveillance. There are two major categories about tracking pedestrians in the mobile surveillance: 1) tracking in a single moving camera: this is a quite challenging task because of the combined effects of egomotion, blur, and rapidly changing lighting conditions, the field of view (FOV) of a camera in this category is usually in front of the vehicle and dynamically changes when the vehicle moves [2], [3]; 2) tracking across multiple moving cameras: this category needs to build upon the

Manuscript received February 27, 2015; revised May 27, 2015; accepted June 27, 2015. Date of publication July 13, 2015; date of current version August 10, 2015. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Haohong Wang.

K.-H. Lee is with the Information Processing Laboratory, Department of Electrical Engineering, University of Washington, Seattle, WA 98105 USA (e-mail: ykhlee@uw.edu).

J.-N. Hwang is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98105 USA (e-mail: hwang@uw.edu).

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

Digital Object Identifier 10.1109/TMM.2015.2455418

reliable results of the tracking in a single moving camera. The challenge is that appearance features extracted from the same pedestrian by different cameras may be inconsistent. There are two scenarios associated with the latter category: overlapping and non-overlapping FOV scenarios. The overlapping FOV scenario is that a pedestrian simultaneously appears in two or more different cameras' FOVs; thus, the re-identification issues can be facilitated by exploiting the cameras' differences based on the overlapping views. The non-overlapping FOV scenario means that a pedestrian enters in a camera's FOV then exits, and later enters in other cameras' FOVs which do not share common region with the previously exited FOV. The problem is more difficult due to dynamic environment, lighting changing and unexpected camera locations. Hence, the methods for tracking under static cameras [13]–[22] cannot be directly applied to the tracking across moving cameras. Moreover, both overlapping and non-overlapping scenarios may occur alternatively during the tracking, thus make this task even more challenging.

In this paper, we deal with the tracking across multiple driving recorders, which can also be extended to other types of moving cameras (such as moving robots or flying drones), and propose a framework to track on-road pedestrians recorded in the videos. We assume a cloud server is used to collect the driving information of the vehicles via a mobile surveillance network. First, pedestrian tracking in a single moving camera is applied to each video. Based on the single-camera-tracking results, we formulate the problem of tracking across cameras as a multi-label classification task, which determines each target belonging to one or several cameras' FOVs by considering association likelihood of the target as calculated based on targets' motion cues and appearance features. When a target is out of camera's FOVs, we predict the target's potential moving trajectory facilitated by an open map service such as *Google Maps*. Moreover, by using the *Google Earth* service, a 3-D visualization of dynamic scene can be reconstructed for users to see a holistic view or different viewing perspectives of the 3-D scenes reconstructed by the multiple videos.

The rest of the paper is organized as follows. Section II gives a brief survey on the related work. In Section III, the overview of the proposed framework is provided. Section IV depicts the methodologies used in the proposed framework. In Section V, pedestrian association likelihood based on appearance features is specifically discussed. The experimental results are shown in Section VI, followed by the conclusion in Section VII.

II. Related Work

The critical issue of pedestrian tracking across multiple cameras is to understand the correlation and association of a moving object among multiple cameras, and then correctly identify and track the object from one camera to another (successful label handoff). Many literatures [4]–[12] treat the problem as person re-identification task, which commonly learns sets of descriptors and/or the metric functions to compute the similarities between the people using a pre-collected dataset. However, the tracking problem across multiple moving cameras is beyond the re-identification task. Since the mapping between two cameras keeps on changing when the cars move, such conditions will prevent the direct use of most person re-identification approaches.

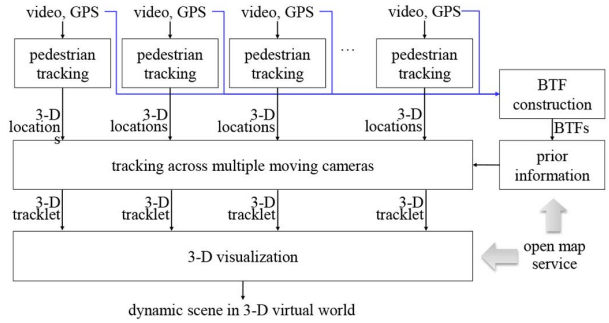


Fig. 2. Overview of the proposed framework.

Tracking across multiple *static* cameras with overlapping views has been well investigated [13]–[15], by exploring the spatial-temporal homography and association of color information between the cameras. In [16], [17], the authors utilize a probabilistic occupancy map to deal with human tracking under multiple cameras with overlapping FOVs. As for non-overlapping scenarios, time-space cues and appearance features are normally used for the label handoff. The problem then becomes how to learn the time-space and appearance relationships between cameras effectively. In addition to the supervised [13], [18] and semi-supervised [19] learning schemes, there are several approaches which effectively use unsupervised learning schemes. In Gilbert's work [20], color information is used in building the transition time distribution. In [21], the authors utilize game theory to solve the label handoff issue in a collaborative PTZ camera network. Recently, Chu and Hwang [22] propose a fully unsupervised approach, which systematically estimates the camera link model between a pair of non-overlapping cameras, in terms of solving a permutation matrix, by considering the transition time distribution along with and the holistic and regional human body color/texture appearance information.

To our best knowledge, there have been very few literatures discussing the issues of tracking across multiple moving cameras. Zou and Tan [23] propose a collaborative Visual Simultaneous Localization And Mapping (V-SLAM) scheme based on the structure from motion (SfM) framework for multiple cameras. These cameras move independently and cooperate with each other, to reconstruct the 3-D trajectories of moving objects in a highly dynamic environment. However, Zou's work focuses on how to collaboratively reconstruct the 3-D objects from the multi-view videos; the tracking problems such as trajectories prediction and association of the objects are not considered.

III. Framework Overview

In this paper, we assume a mobile surveillance network where a server collects the driving information of multiple vehicles within a local area and a period of time $t = 1, \ldots, T$. The driving information includes (intrinsic) camera parameters, GPS, global timestamp and videos, which are well synchronized by the global timestamps. Fig. 2 shows the overview of the proposed framework. First, pedestrian tracking, which produces the moving trajectory and associated features of

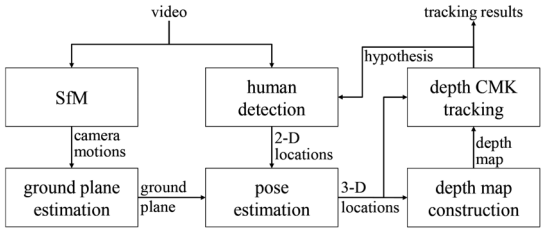
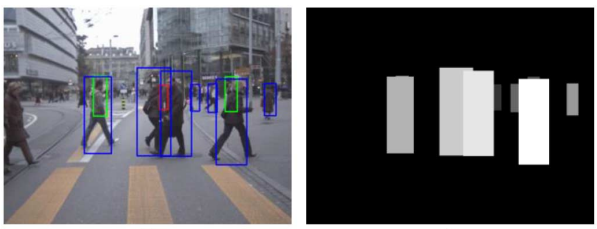


Fig. 3. Single-camera-tracking system.



(a) (b)

Fig. 4. Example of the depth map, showing (a) detections and (b) depth map, where higher intensity indicates detected pedestrians are closer to the camera.

the tracked person (tracklet) in 3-D space, in a single camera is applied to each video. The videos are then used to build Brightness Transfer Functions (BTFs) for compensating color diversity of the cameras. After estimating the pedestrians' 3-D tracklets in each camera, the pedestrian tracking across multiple cameras is further applied, facilitated with the BTFs and map prior. Finally, the tracked pedestrians' 3-D tracklets are summarized and 3-D visualized in the 3-D real-world environment.

A. Tracking Under a Single Moving Camera

The pedestrian tracking under a single moving camera is based on the tracking-by-detection method proposed in our previous work [24], which separately tracks pedestrians in a single moving camera. Fig. 3 shows the overview of the single-camera-tracking system. The method starts with pedestrian detection by a human detector to locate the pedestrians in 2-D frames. Then, camera motions can be well calibrated by the V-SLAM framework. Facilitated by the camera motions and ground plane estimation, the 3-D locations of the pedestrians can be accurately estimated. By taking 3-D information (depth) into account to avoid occlusion issues, the method extends the constrained multiple-kernel (CMK) tracking approach [25] to effectively optimize the detected target association between consecutive frames. Fig. 4(a) shows the result of pedestrian detection and Fig. 4(b) is the corresponding depth map. As shown in Fig. 4(a), two green-boxed human targets are almost occluded by the other targets in front of them, while a red-boxed target is totally occluded by other targets. Such occlusion issues can be effectively resolved if the tracking is handled in 3-D space.

Once the pedestrians are successfully tracked in each moving camera, the profile of the tracked targets, including motion cues (i.e., the pedestrians' 3-D (GPS) locations can be inferred from the camera's GPS location) and appearance features, can be obtained.

B. Tracking Across Multiple Moving Cameras

Firstly, we describe the notations used in the proposed algorithm. Given N tracked targets in total by M moving cameras, the actual number of the distinct tracked pedestrians N' should be smaller than or equal to the total number tracked in the M cameras, i.e., $N' \leq N = \Sigma_{j=1}^{M} N_j$, where there are N_j targets tracked in the jth moving camera, since one pedestrian can possibly appear in more than one camera's FOV. For the ith tracked target, $i = 1, \ldots, N$, the target profile O_i^t at timestamp t is composed of its appearance model and motion model. For the jth moving camera, $j = 1, \ldots, M$, the camera profile C_j^t at t is composed of calibrated camera parameters $\mathbf{P}(C_j^t)$, GPS locations $gps(C_j^t)$, forwarding directions $dir(C_j^t)$, and a set of target profiles $\mathbf{O}(C_j^t)$, which are specifically derived profiles for all the targets appearing in the jth camera's FOV. A tracklet l_i is a set which has the ith target profiles for all possible t. $\mathbf{C}_\times$ is a set of camera profiles, containing the predicted intersections of the specific target and the cameras' FOVs at a later t_j. Finally, p_{map} is a prior of map topology which describes routing information.

For each t (i.e., current timestamp), we check whether each tracked target is exiting camera's FOV or not. If the ith target is exiting the jth camera's FOV, this implies the target either disappears forever or later enters into other cameras' FOVs, we then apply three steps to possibly determine its subsequent locations. 1) *Prediction*: By utilizing the p_{map}, we can predict when the ith target possibly intersects with the jth camera's FOV at a later t_j, as shown in Fig. 5(a). These intersections in terms of camera profiles are put into $\mathbf{C}_\times$. 2) *Classification*: Based on the appearance features and motion cues, we associate the exiting target O_i^t with every target which appears in the possible intersections (i.e., the elements of $\mathbf{C}_\times$), as shown in Fig. 5(b). 3) *Interpolation*: If the exiting target is determined to appear in some cameras' FOVs, we uniformly interpolate the hypothesized target profiles into the tracklet of the associated target(s), as shown in Fig. 5(c). On the other hand, if the ith target appears in the jth camera's FOV, we associate O_i^t with the targets in one or multiple cameras by classification operation based on the previous target profile O_i^{t-1}. Moreover, if the target appears in multiple cameras simultaneously, we apply the *Overlapping* operation which bundle-adjusts the targets 3-D locations from multiple views, as shown in Fig. 5(d). Fig. 6 shows an example of the proposed framework, where each row represents a tracklet with horizontal axis being the time line. In the example, O_1^4 is classified to camera 1 and 3 at $t = 4$, so the overlapping operation is applied to O_1^4 and O_9^4; when $t = 8$, O_2^8 is predicted to appear in C_2^{10} and C_3^{21}, i.e., $\mathbf{C}_\times = \{C_2^{10}, C_3^{21}\}$. When $t = 10$, there is no possible candidate for O_2^8. At $t = 21$, the targets in camera 3 are classified based on O_2^8, while O_8^{21} is regarded as a candidate. Hence, the hypothesized target profiles uniformly interpolated from O_2^8 to O_8^{21} are inserted into l_8.

IV. Methodologies

In this section, we describe the details of the operations mentioned above: prediction, classification, interpolation, and overlapping.

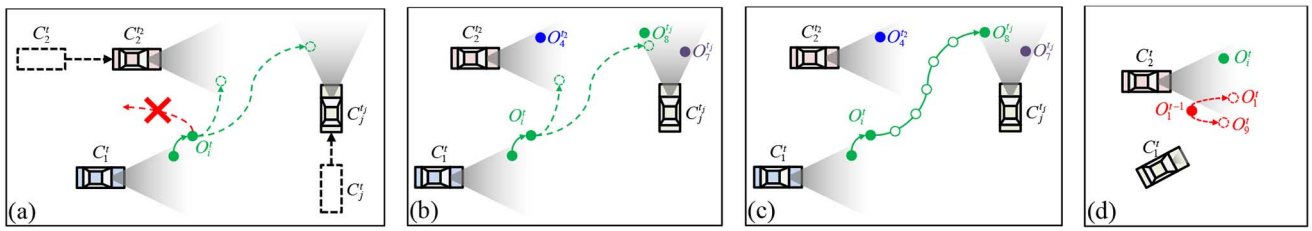


Fig. 5. Four operations in the proposed framework. (a) Prediction: O_i^t is predicted to appear in the camera 2 at t_2 and the camera j and t_j. (b) Classification: only $O_8^{t_j}$ is associated with O_i^t. (c) Interpolation: insert hypothesized target profiles (non-solid circles) from O_i^t to $O_8^{t_j}$ along the route provided by p_{map}. (d) Overlapping: If a target associate with targets in more than one camera's FOV, we apply bundle adjustment to the targets.

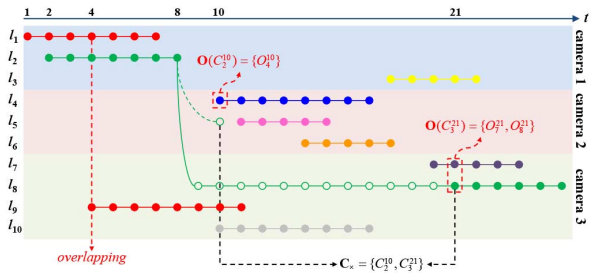


Fig. 6. Example of the proposed framework in case of $M = 3$ and $N = 10$ ($N' = 8$ in this case), where each point is a target profile, l_1 and l_9 are identical, l_2 and l_8 are identical. The non-solid circles are the hypothesized target profiles.

A. Prediction

In the prediction, p_{map} is used to denote the topology information (i.e., routing path, transition time, and direction) with respect to GPS location. Based on p_{map}, a shortest route from one GPS location (gps_1) to another (gps_2) can be estimated, so that the transition time $t_s(gps_1, gps_2)$, and the forwarding direction of the route at gps_1, denoted by $dir_{map}(gps_1|gps_2) \in \mathbb{R}^2$, can be obtained. In this paper, we adopt Google Maps service to generate p_{map}, by querying routing information between two GPS locations.

The prediction is to find the possible intersections of the ith target and the jth camera's FOV in the near future. Given an O_i^t, in order to locate an intersection with the jth camera, we try to find a later timestamp t_j, such that the transition time from $gps(O_i^t)$ to the FOV of $C_j^{t_j}$, denoted by t_o, is similar to the transition time from the $gps(C_j^t)$ to the $gps(C_j^{t_j})$, denoted by t_c. Consider that $gps(C_j^{t_j}, r_{fov}|O_i^t)$ is a GPS location which has the shortest route from $gps(O_i^t)$ to $C_j^{t_j}$'s FOV with the visible range r_{fov}, then t_o and t_c can be defined as

$$\begin{cases} t_o = t_s\left(gps(O_i^t), gps(C_j^{t_j}, r_{fov}|O_i^t)\right), \\ t_c = t_s\left(gps(C_j^t), gps(C_j^{t_j})\right). \end{cases} \tag{1}$$

Moreover, the estimated target's forwarding direction $dir(\bullet)$, as obtained from the results of single-camera-tracking, and the measured target's forwarding direction $dir_{map}(\bullet)$, as obtained from p_{map}, should be consistent, as shown by the red-dot line in Fig. 5(a). This results in a constraint to the selection of $C_j^{t_j}$

$$\begin{aligned} &\mathbf{C}_\times = \left\{ C_j^{t_j} \,||t_o - t_c| < \tau_t \right\} \\ &\text{s.t.} \quad dir(O_i^t) \cdot dir_{map}\left(gps(O_i^t)|gps(C_j^{t_j}, r_{fov}|O_i^t)\right) < \tau_{dir} \end{aligned} \tag{2}$$

where τ_{dir} and τ_t are the thresholds to restrict the directions and transition time, respectively.

B. Classification

A tracked target may appear in one or several cameras' FOVs either at the same time or at subsequent timestamps. Without loss of generality, we assume the task is to associate one tracked target with the other at a (later) different timestamp. Therefore, we treat this scenario as a multi-label classification task. To measure the likelihood of the presence of the ith previously tracked target in the jth camera's FOV at t, we calculate the association likelihood based on the targets' appearance and motion information respectively. Given an O_i^t to be classified and a reference profile O_i^{ref}, the association likelihood of the ith target in the jth camera's FOV is defined as

$$\ell_{i,j}^t = \max\left\{ \ell_{app}(O_i^{ref}, O_k^t) \cdot \ell_{mo}(O_i^t, O_k^t) | \forall O_k^t \in \mathbf{O}(C_j^t) \right\} \tag{3}$$

where ℓ_{app} is the likelihood of the targets' matching of appearance (see Section V); ℓ_{mo} is the likelihood of the ith target appearing in the jth camera's FOV, and is defined as a Gaussian weighted function of the distance between two targets

$$\ell_{mo}(O_i^t, O_k^t) = \rho \cdot G_{0,\sigma_{mo}}\left(X(O_i^t) - X(O_k^t)\right) \tag{4}$$

where $G_{0,\sigma_{mo}}(\bullet)$ is the a standard Gaussian function with $\mu = 0$ and $\sigma = \sigma_{mo}$, ρ is a normalization value, and $X(\bullet)$ is the 3-D location of the assigned target profile.

Thus, regarding each camera's FOV as a class, we tend to classify the ith target at each t, by formulizing the problem as a Quadratic Boolean Problem (QBP)

$$\max_{\mathbf{v}_i^t} (\mathbf{v}_i^t)^{\mathrm{T}} \mathbf{L}_i^t (\mathbf{v}_i^t), \mathbf{L}_i^t = \begin{bmatrix} l_{11}^t & \cdots & l_{1M}^t \\ \vdots & \ddots & \vdots \\ l_{M1}^t & \cdots & l_{MM}^t \end{bmatrix} \tag{5}$$

where $\mathbf{v}_i^t = [v_1, \ldots, v_M]^T$ is a vector of indicator variables such that $v_j = 1$ if the target appears in the jth camera's FOV, and $v_j = 0$ otherwise. The diagonal elements of $\mathbf{L}_i^t$ are $\ell_{i,p}^t$, as defined in (3), i.e., $l_{pp}^t = \ell_{i,p}^t$, denoting the association likelihood of the ith target appearing in the pth cameras' FOVs; while the rest of the elements are the association likelihood of the targets which correspond to the ith target in the pth and the qth cameras' FOVs separately, i.e., $l_{pq}^t = \ell_{i,p}^t \cdot \ell_{i,q}^t - \eta, p \neq q$; η is the penalty for non-correspondence of two targets. Such a problem can be solved by the standard optimization methods [26].

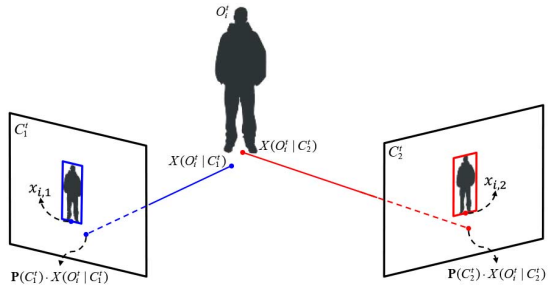


Fig. 7. Illustration of the bundle adjustment in the overlapping operation, where the estimated 3-D location will be optimized by minimizing the reprojection error.

In general, the dimension of $\mathbf{v}_i^t$ is M (the total number of the cameras). However, larger M may make $\mathbf{L}_i^t$ a sparse matrix, resulting in divergence of the problem and higher computational cost. To lower the dimension of $\mathbf{v}_i^t$, we only select the cameras whose 3-D locations are within the visible range r_{fov}

$$\forall C_k^t \quad \text{s.t.} \quad \left\| X(O_i^t) - X(C_k^t) \right\|_2 < r_{fov} \tag{6}$$

where $X(\bullet)$ is the 3-D locations of the specific profile.

C. Interpolation

Once the prediction of the ith tracked target intersecting with the jth camera at a later timestamp t_j is confirmed, we insert the hypothesized target profiles $O_i^{t'}$ into l_i, from t to t_j. All $O_i^{t'}$ are constructed by the appearance models of O_i^t, and their 3-D locations (as well as GPS locations) are uniformly interpolated along the routes, as provided by Google Maps service. If the predicted target intersects with multiple cameras' FOVs, implying that the disappeared (exited) target will later enter into two or more cameras' FOVs at different t_j. Therefore, multiple hypothesized tracklets are inserted into the corresponding l_i.

D. Overlapping

If O_i^t appears in multiple cameras' FOVs at t, we apply a standard bundle adjustment formulation to optimize the 3-D locations of the O_i^t in multiple cameras' FOVs. The set of 3-D points and the corresponding 2-D points are used in the bundle adjustment [27] process to iteratively minimize the total reprojection error

$$\hat{X}(O_i^t) = \underset{X(O_i^t)}{\arg\min} \sum_{C_j^t \in \mathbf{C}_*} d_{proj}\left(\mathbf{P}(C_j^t) \cdot X(O_i^t|C_j^t), x_{i,j}\right) \tag{7}$$

where $x_{i,j}$ is the observed 2-D location (i.e., middle of bottom of the pedestrian blob) corresponding to $X(O_i^t|C_j^t)$ from the jth camera's FOV, $\mathbf{P}(C_j^t)$ is the projective matrix of the jth camera at t, $\mathbf{C}_*$ is a set of C_j^t which all include $O_i^t \in \mathbf{O}(C_j^t)$ for all j, and $d_{proj}(\bullet)$ is the distance measurement between the reprojected locations and the observed locations in the image. Such a nonlinear least-square problem is solved by the Levenberg-Marquardt algorithm. Fig. 7. simply illustrates the idea of the bundle adjustment in the overlapping operation.

Unlike the classic multi-view stereo algorithm, where the bundle adjustment is used to globally optimize camera motions and target's 3-D location, we maintain the camera motions to optimize the target's 3-D location for system stability and reliability.

V. Pedestrian Association Likelihood

In this section, we describe the details of the $\ell_{app}(\bullet,\bullet)$, i.e., the association likelihood of two targets according to their appearance features. The appearance features, such as color and texture, are commonly used to represent the targets. However, due to viewing perspectives and diversity of camera devices, a target's color-channel image intensities extracted from one camera are normally different from those of the other camera. Therefore, Brightness Transfer Function (BTF) [13] between two cameras is necessarily applied before the feature matching.

A. Brightness Transfer Functions

Generally, the BTFs can be estimated through overlapping FOVs of cameras. Our assumption is that the lighting condition in each camera is roughly consistent within a short period of time in the vicinity. Based on the assumption, we utilize "spatially" overlapping cameras' FOVs, i.e., a camera's FOV overlaps with the other cameras' FOVs, but not necessary to be simultaneous. To calculate the BTF between two cameras, we first group the camera views (i.e., video frames) by applying mean-shift clustering to the GPS locations of the cameras. If a cluster includes more than one camera, the cluster is regarded as overlapping. Hence, for any two cameras within an overlapping cluster G_g, we can have a set $\mathbf{F}_g^{p,q}$ containing pairs of (C_p^t, $C_q^{t'}$), where $gps(C_p^t)$ is the nearest location to $gps(C_q^{t'})$

$$\mathbf{F}_g^{p,q} = \left\{ (C_p^t, C_q^{t'}) | \min\left(d_{phy}\left(gps(C_p^t), gps(C_q^{t'})\right)\right)\right\} \tag{8}$$

where $d_{phy}(\bullet)$ is the physical distance between two GPS locations. Therefore, the SIFT-feature matching is used to obtain all the pairs of the matching points, which are then used to estimate the BTF from the pth camera to the qth camera within the gth cluster, denoted by $f_{BT,g}^{p\to q}$. Based on the color histograms calculated by the matching points, we then apply the RANSAC algorithm to obtain the optimal $f_{BT,g}^{p\to q}$.

On the other hand, if there is no spatially overlapping cameras' FOVs, we apply identity matrix to the BTF, i.e., $f_{BT,g}^{p\to q} = \mathbf{I}$. This issue will be further discussed in Section VI-D.

B. Appearance Features

To evaluate the association likelihood of the pedestrians, the proposed framework adopts appearance (low-level) features. Many appearance features have been developed and have shown good performance in different applications. Color information is widely used and is considered to generate high-impact features in general cases [5]–[7], [10], [28]. Some approaches additionally consider shape [10], [11] and texture [7]–[9], [29], [30], to improve the performance in case of large color variations. Recently, many patch-based/local features [5]–[10] have attracted many attentions for the effective use of spatial information. From pedestrian images, lots of local patches are extracted to describe regional features, which have the advantage of invariance to misalignment, pose variation, and the change in viewpoint.

Assume the appearance features of a pedestrian are A_κ, $\kappa = 1, 2, \ldots, K$, and the κth appearance feature can be

extracted from the object profile, which is pre-applied the corresponding BTF, i.e., $A_\kappa(f_{BT,g}^{p\to q}(O_i^t))$, where $f_{BT,g}^{p\to q}$ is the BTF from the pth camera to the qth camera and O_i^t is within the gth cluster. If the gth cluster is not overlapping (i.e., no overlapping FOVs are grouped), the BTF, from the pth camera to the qth camera, closest to the gth cluster is selected. Hence, the association likelihood of the κth appearance features, denoted by $\ell_{app,\kappa}(A_\kappa, A'_\kappa)$, is defined as the similarity between A_κ and A'_κ. By taking multiple appearance features into account, a total likelihood for the feature matching is defined by the linear combination of the selected appearance features

$$\begin{aligned}\ell_{app} &= w_1 \cdot \ell_{app,1} + w_2 \cdot \ell_{app,2} + \cdots + w_K \cdot \ell_{app,\kappa} + b \\ &= \mathbf{w} \cdot \mathbf{l}_{app} + b \end{aligned} \tag{9}$$

where $\mathbf{w} \in \mathbb{R}^\kappa$ consists the weights for the appearance features and represents the impact factor to the corresponding features; $\mathbf{l}_{app} \in \mathbb{R}^\kappa$ is a vector of the association likelihoods; and b is the offset of hyperplane.

By integrating some well-developed features, the proposed framework can take advantage of several appearance features' properties, to achieve nice performance when tracking across multiple moving cameras. Nevertheless, some features are mutually complementary and some are correlated with each other. Hence, how to select useful features is quite important for the proposed framework. The appearance feature selection is discussed in Section VI-A.

C. Interior Training

To determine the corresponding weights for the selected appearance features [i.e., $\mathbf{w}$ in (9)], a training stage is necessary. However, FOV dynamically changes when a camera is moving, there is no chance to label the ground truth of the corresponding targets. To solve this problem, we apply an interior training scheme to determine $\mathbf{w}$. The idea is to label positive and negative training data sets within the target profiles, based on their spatial-temporal relationship. More specifically, two target profiles belonging to the same tracklet are labeled as positive because they have been successively tracked in a single moving camera. On the contrary, if two targets appear in one camera's view simultaneously, these two target profiles are impossible to be the same target, i.e., ℓ_{mo} is smaller than a threshold. Hence, these two target profiles are labeled as negative. Next, we use a standard linear Support Vector Machine (SVM) to obtain the $\mathbf{w}$, so as to be used in (9). Finally, to represent the output of SVM as likelihood, we apply a sigmoid function to normalize ℓ_{app}[31].

VI. Experimental Results

Several experiments are conducted to demonstrate the performance of our proposed framework. To our best survey, there is no public dataset specific for the case of multiple moving cameras based on driving recorders. Hence, we try our best to record the videos and build the test datasets. The configurations of the devices and datasets are shown in Table I, and the recorded videos are pre-synchronized. To evaluate the tracking performance, we consider the following metrics which are widely used in the previous work [2], [24], [32].

TABLE I
Configurations of the Devices and Datasets

Device	Type	Resolution	fps
1	PAPAGO P2	1280×720	30
2	Pro V DV-2021	640×480	30
3	DOD F500LHD	1280×720	30
4	PAPAGO P2	1280×720	30

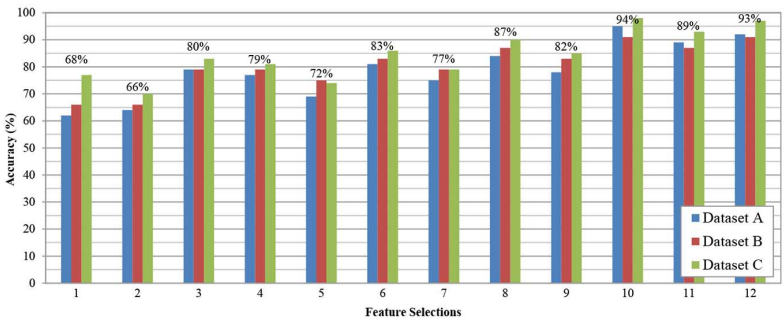
Dataset	Devices	*T*	Detection rate (%)	False Positive Per Image (FPPI)
A	1,2	4015	64.78	0.244
B	1,2	2042	79.33	0.127
C	1,2,3,4	4169	75.11	0.103

- Most tracked tracklets (MT): the number of tracklets that successfully tracked more than 80% frames across all video sequences.
- Partially tracked tracklets (PT): the number of tracklets that successfully tracked between 20% and 80%.
- Most lost trajectories (ML): the number of tracklets that successfully tracked less than 20%.
- Fragmentation (FM): the number of times a tracklet is interrupted. Furthermore, we evaluate the FM in each single camera (sFM) and across multiple cameras (mFM), respectively.
- ID switches (IDS): the number of times two tracklets switch their IDs, in a single camera (sIDS) and across multiple cameras (mIDS).

A. Appearance Features Selection

The proposed framework can apply multiple appearance features for calculating pedestrian association likelihood. To select proper appearance features, we tested many combinations of different appearance features to reach the conclusion of the most effective features for the proposed framework. We test the selected appearance features with three datasets, and measure the performance in terms of average accuracy of the pedestrian matching. Moreover, since background subtraction cannot be applied in the moving camera scenarios, no explicit segmentation masks can be created, we thus create an ellipse bounding mask for each tracked target when calculating the appearance features.

The features to be selected include color information such as weighted color histogram (WCH) and maximally stable color regions (MSCR) [28], texture information such as scale invariant feature transform (SIFT) [29] and local binary pattern (LBP) [30], and the patch-based/local descriptors such as recurrent high-structured patches (RHSP) [5]. Firstly, we select color information since it is widely used and is considered as a high-impact feature. From Fig. 8, the average accuracy of using WCH is 68%, and the accuracy of using MSCR is 66%; which implies color information performs sufficiently well for the matching in our scenarios. Then, we further add texture features by testing the selections no.3 (WCH+LBP) and no.4 (WCH+SIFT), and the average accuracy are increased to 80% and 79%, respectively. This implies that texture can further improve the matching performance when combined with the WCH. Next, we consider the patch-based features by testing the RHSP combined with the color and texture features. As shown in the figure, the average accuracy of using the selection no.5 (WCH+RHSP) is 72%. The accuracy of using the selection



No.	Feature Selections	Segmentations
1	WCH	NO
2	MSCR	NO
3	WCH+LBP	NO
4	WCH+SIFT	NO
5	WCH+RHSP	NO
6	WCH+RHSP	YES
7	WCH+LBP+RHSP	NO
8	WCH+LBP+RHSP	YES
9	WCH+MSCR+RHSP (**SDALF**)	NO
10	WCH+MSCR+RHSP (**SDALF**)	YES
11	WCH+MSCR+SIFT	NO
12	WCH+MSCR+LBP	NO

WCH: Weighted Color Histogram, **MSCR**: Maximally Stable Color Regions [28], **LBP**: Local Binary Pattern, **SIFT**: Scale-Invariant Feature Transform [29], **RHSP**: Recurrent High-Structured Patches [5], **SDALF**: Symmetry-Driven Accumulation of Local Features [5].

Fig. 8. Comparison of matching accuracy of different selections.

(a) (b)

Fig. 9. Visual tracking results in Dataset A, where the top rows are the recorded frames, and bottom rows are the corresponding 3-D visualization. (a) Frames from device 1 at $t = 1128$ (left), $t = 2154$ (middle), and $t = 2984$ (right). (b) Frames from device 3, at $t = 2477$ (left), $t = 2977$ (middle), and $t = 3266$ (right).

(a) (b)

Fig. 10. Visual tracking results in Dataset B, where the top rows are the recorded frames, and bottom rows are the corresponding 3-D visualization. (a) Frames from device 1 at $t = 89$ (left), $t = 1070$ (middle), and $t = 1497$ (right). (b) Frames from device 3, at $t = 662$ (left), $t = 1220$, and $t = 1497$ (right).

no.7 (WCH+LBP+RHSP) is 77%, which is worse than that of using the selection no.3 (WCH+LBP). This means that the RHSP cannot improve the matching performance in our scenarios. We also test the selection no.9 (WCH+MSCR+ RHSP), which is known as the symmetry- driven accumulation of local features (SDALF) [5]; however, the result (82%) is still not impressive. This is because such patch-based descriptors highly rely on accurate segmentations of the pedestrian blobs, which cannot be easily achieved in the moving cameras. Finally, we recall the MSCR, which contains both color information and regional description. As shown in the results, the selection no.11 (WCH+MSCR+SIFT) can achieve average accuracy of 89% and the selection no.12 (WCH+MSCR+LBP) can reach 93%, both can achieve much better performance than other selections. This means that global features (WCH and LBP) mainly contribute the results, and regional features (MSCR and SIFT) are necessary to enhance the performance. Finally, to better understand the influence of segmentation performance, given the segmentations of the pedestrian blobs which are manually refined, we test the selections which have RHSP. The results show that the performance of the RHSP has significant improvement when the manual segmentations are used. The average accuracy of the selection no.6 (WCH+RHSP) is 83%, the selection no.8 (WCH+LBP +RHSP) is 87%, and the SDALF is up to 94%. This demonstrates that the well-developed features can be applied to the proposed framework under proper configurations, to obtain better performance.

In summary, the selected features should include global features such as color and texture information. Moreover, regional features are also necessary for improvement of the tracking performance. The patch-based features can be taken into account if accurate segmentations of the pedestrians' blobs are available.

B. Tracking Results

We test three datasets (A, B, and C), with different scenarios, to demonstrate the performance of the proposed method. In the dataset A, a pedestrian in one camera's FOV will leave for a while, and then enter into another (or the same) camera's FOV. Fig. 9. shows that the pedestrians appearing in the camera 1's FOV, will later appear in the camera 3's FOV. For example,

(a) (b)

Fig. 11. Visual tracking results in Dataset C, where the top rows are the recorded frames, and bottom rows are the corresponding 3-D visualization. (a) Non-overlapping case, tracking from device 1 at $t = 348$ (left), followed by device 3 at $t = 846$ (middle), and device 4 at $t = 3231$ (right). (b) Overlapping case, tracking from device 3 at $t = 902$ (left), and device 1's view (middle) overlapped with device 3 (right), both at $t = 1175$.

the pedestrian no.46 at $t = 1128$ in Fig. 9(a) will appear at $t = 2477$ in Fig. 9(b). In the dataset B, a pedestrian in one camera's FOV, will also enter into another camera's FOV at the same time. As shown in Fig. 10, the pedestrian no.3, no.7, and no.18 appear in both camera 1's and camera 3's FOVs almost at the same time. In the dataset C, we simultaneously recorded four videos by four driving recorders; this complicated scene includes all the scenarios mentioned in the previous section. For example, Fig. 11(a) shows a 2-cameras overlapping case, while Fig. 11(b) shows a non-overlapping case. For single-camera-tracking in the proposed framework, we adopt the constrained 2-kernel based method in [24], and compare with the flow network based association method in [33].[1] As for tracking across multiple cameras, we compare the proposed approach with the MvsM scheme, which is widely used for person re-identification [4], [5].

The tracking results are shown in Table II, where GT denotes the number of trajectories in the ground truth. The lower FM and IDS values represent the better ability to track pedestrians successively. From the table, the methods with C2 K perform better than that with FN, even with different feature selections (no.10, no.11, and no.12 in our experiments). This implies that better single-camera-tracking method provides better tracking results for the multiple-cameras-tracking framework. Next, we compare the proposed approach with the MvsM scheme, based on several feature selections. As shown in the results, the proposed approach has lower mFM and mIDS than the MvsM by using feature selections no.10 and no.12. This is because the MvsM identifies the pedestrians by only considering appearance features; but the proposed method further utilizes the relative 3-D locations predicted by the map prior. Moreover, with the same feature selections, the results of the proposed approach are better than that of the MvsM. This indicates that the proposed approach can improve the performance by adopting well-developed features. These results clearly show favorable performance of the proposed framework, not only in overlapping and non-overlapping scenarios, but also in complicatedly mixed scenarios.

Based on their relative 3-D (GPS) locations, we create a 3-D visualization of the dynamic scene, so as to observe what happens to the roads/streets from different aspects of view, as shown in Figs. 9–11. The 3-D real-world environment is built upon the Google Earth, where the pedestrians are replaced by an avatar-like 3-D human model, and the vehicles equipped with cameras are also represented by the default 3-D vehicle models. The videos associated with the simulations and the demo of 3-D visualization can be viewed in our website.[2]

TABLE II
COMPARISON OF THE TRACKING RESULTS

Dataset A										
Single	**Multiple**	**Features**	**GT**	**MT**	**PT**	**ML**	**sFM**	**sIDS**	**mFM**	**mIDS**
FN	Proposed	no.11	71	27	13	2	16	3	7	5
FN	Proposed	no.12	71	27	13	2	16	3	7	5
FN	Proposed	no.10	71	27	13	2	16	3	5	3
C2K	Proposed	no.11	71	34	8	4	9	1	6	3
C2K	*MvsM*	no.9	71	34	8	4	9	1	7	9
C2K	Proposed	no.12	71	34	8	4	9	1	**5**	**2**
C2K	*MvsM*	no.12	71	34	8	4	9	1	7	6
C2K	Proposed	no.10	71	34	8	4	9	1	**3**	**2**
C2K	*MvsM*	no.10	71	34	8	4	9	1	5	4

Dataset B										
Single	**Multiple**	**Features**	**GT**	**MT**	**PT**	**ML**	**sFM**	**sIDS**	**mFM**	**mIDS**
FN	Proposed	no.11	12	12	9	1	1	1	3	2
FN	Proposed	no.12	12	12	9	1	1	1	2	2
FN	Proposed	no.10	12	12	9	1	1	1	1	2
C2K	Proposed	no.11	12	10	1	0	1	0	2	2
C2K	*MvsM*	no.9	12	10	1	0	1	0	2	3
C2K	Proposed	no.12	12	10	1	0	1	0	**1**	**0**
C2K	*MvsM*	no.12	12	10	1	0	1	0	2	3
C2K	Proposed	no.10	12	10	1	0	1	0	**1**	**0**
C2K	*MvsM*	no.10	12	10	1	0	1	0	2	2

Dataset C										
Single	**Multiple**	**Features**	**GT**	**MT**	**PT**	**ML**	**sFM**	**sIDS**	**mFM**	**mIDS**
FN	Proposed	no.11	43	23	12	6	4	2	8	5
FN	Proposed	no.12	43	23	12	6	4	2	7	5
FN	Proposed	no.10	43	23	12	6	4	2	3	4
C2K	Proposed	no.11	43	27	10	4	0	0	4	1
C2K	*MvsM*	no.9	43	27	10	4	0	0	4	13
C2K	Proposed	no.12	43	27	10	4	0	0	**1**	**1**
C2K	*MvsM*	no.12	43	27	10	4	0	0	4	6
C2K	Proposed	no.10	43	27	10	4	0	0	**1**	**0**
C2K	*MvsM*	no.10	43	27	10	4	0	0	4	6

FN: flow network based [33], C2 K: constrained 2-kernel based [24], GT: ground truth.

C. Impact of σ_{mo}

The parameter σ_{mo} in (4) plays an important role in our proposed framework, since it not only determines the valid range of the cameras' FOVs, but also filters out the targets whose locations are improper. Fig. 12 shows the results of the mFM and mIDS versus different σ_{mo} in terms of meters. As we can

[1][Online]. Available: http://people.csail.mit.edu/hpirsiav/

[2][Online]. Available: http://allison.ee.washington.edu/kuanhuilee/mmcv

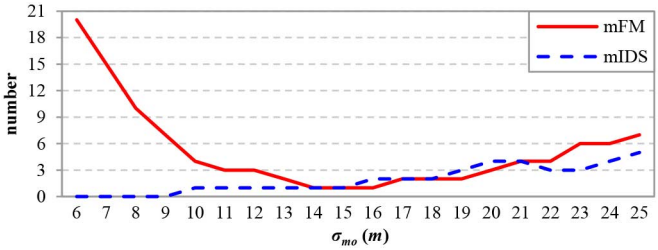


Fig. 12. Results of mFM and mIDS with different σ_{mo}.

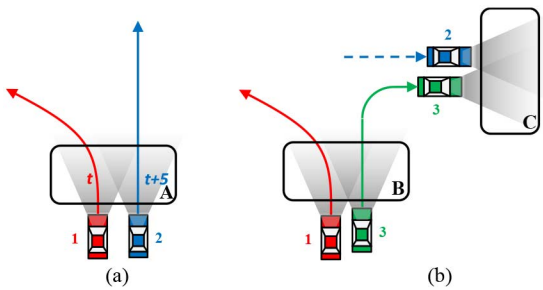


Fig. 13. Examples of spatially overlapping FOVs. (a) FOV1 and FOV2 are spatially overlapping in the region A at different timestamp. (b) FOV1 and FOV2 are non-overlapping; but FOV1 and FOV3 are spatially overlapping in the region B, FOV2 and FOV3 are spatially overlapping in the region C.

observe, the mFM and the mIDS become larger when σ_{mo} is larger. This is because a larger σ_{mo} relaxes the restriction provided by the motion cues, resulting in more failure of the pedestrians' association, as well as mFM. In this case, the scenario is similar to the *MvsM*; that is the association only considering appearance features without taking into account the motion cues (obtained from p_{map}, i.e., open map service), resulting in a typical re-identification task. However, if σ_{mo} is smaller, the mFM becomes larger since the restriction to a pedestrian's presence is too strong to preserve the candidates. This implies that the performance is highly improved when the motion cues are adequately incorporated. Hence, we empirically choose $\sigma_{mo} = 15$ in our experiments.

D. Discussion and Limitations

In the proposed framework, the BTF is calculated by using spatially overlapping FOVs, as shown in Fig. 13(a). If there is no spatially overlapping FOV, as the FOV1 and FOV2 in Fig. 13(b), we apply identity matrix to the BTF, and expect degraded performance. However, thanks to the mobility of the moving cameras, there is still an opportunity to estimate the BTF between two non-overlapped cameras' FOVs. As shown in Fig. 13(b), if FOV1 overlaps with FOV3 in the region B, and FOV3 overlaps with FOV2 in the region C, we can estimate the BTF between FOV1 and FOV2 via FOV3. In the future, we intend to build BTF connectivity between all the collected videos, so that we can explore the relationship between the cameras, and further estimate the BTFs within the connectivity.

Furthermore, the proposed framework can be scaled up to larger systems with larger amounts of moving cameras. Each local region within a time section can be regarded as a processing unit, and the proposed framework is adopted in the unit. Eventually, facilitated by the cloud computing, all the units can be processed in parallel, so as to efficiently construct larger and wider mobile surveillance.

VII. Conclusion

We propose a new framework for tracking pedestrians across multiple moving cameras, by treating the problem as multi-label classification at each timestamp. Based on the appearance and motion cues, facilitated by the Google Maps, the proposed framework evaluates the association likelihood of the targets, so as to associate with the targets across the multiple moving cameras. Moreover, based on their relative 3-D locations, we reconstruct the 3-D visualization of the dynamic scene. In the future, the proposed framework can also be further applied to other classes of on-road objects if the corresponding detectors are available.

References

[1] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1239–1258, Jul. 2010.

[2] A. Ess, B. Leibe, K. Schindler, and L. VanGool, "Robust multiperson tracking from a mobile platform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1831–1846, Oct. 2009.

[3] B. Leibe, K. Schindler, N. Cornelis, and L. VanGool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1683–1698, Oct. 2008.

[4] S. Gong, M. Cristani, S. Yan, and C. C. Loy, *Person Re-Identification*. New York, NY, USA: Springer, 2014.

[5] M. Farenzenz, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*., Jun. 2010, pp. 2360–2367.

[6] W. Li and X. Wang, "Locally aligned feature transforms across views," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 3594–3601.

[7] F. Jurie and A. Mignon, "PCCA: A new approach for distance learning from sparse pairwise constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2666–2672.

[8] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 262–275.

[9] M. Hirzer, P. M. Roth, and M. Kostinger, "Relaxed pairwise learned metric for person Re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 780–793.

[10] W. Zheng, S. Gong, and T. Xiang, "Person Re-identification by probabilistic relative distance comparison," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 649–656.

[11] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu, "Shape and appearance context modeling," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[12] A. Alahi, P. Vandergheynst, M. Bierlaire, and M. Kunt, "Cascade of descriptors to detect and track objects across any network of cameras," *Comput. Vis. Image Understanding*, vol. 114, no. 6, pp. 624–640, 2010.

[13] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *Proc. IEEE Conf. Comput. Vis.*, Oct. 2003, pp. 952–957.

[14] B. Moller, T. Plotz, and G. A. Flink, "Calibration-free camera handover for fast and reliable person tracking in multi-camera setups," in *Proc. IEEE Conf. Pattern Recog.*, Dec. 2008, pp. 1–4.

[15] C.-T. Chu, J.-N. Hwang, K.-M. Lan, and S.-Z. Wang, "Tracking across multiple cameras with overlapping views based on brightness and tangent transfer functions," in *Proc. ACM/IEEE Int. Conf. Distrib. Smart Cameras*, Aug. 2011, pp. 1–6.

[16] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 267–282, Feb. 2008.

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

[18] T. D'Orazio, P. Mazzeo, and P. Spagnolo, "Color brightness transfer function evaluation for non overlapping multi camera tracking," in *Proc. ACM/IEEE Int. Conf. Distrib. Smart Cameras*, Sep. 2009, pp. 25–32.
[19] S.-I. Yu, Y. Yang, and A. Hauptmann, "Harry Potter's Marauder's Map: Localizing and tracking multiple persons-of-interest by nonnegative discretization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 3714–3720.
[20] A. Gilbert and R. Bowden, "Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 125–136.
[21] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, "Collaborative sensing in a distributed PTZ camera network," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3282–3295, Jul. 2011.
[22] C.-T. Chu and J.-N. Hwang, "Fully unsupervised learning of camera link models for tracking humans across non-overlapping cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 979–994, Jun. 2014.
[23] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.
[24] K.-H. Lee, J.-N. Hwang, G. Okopal, and J. Pitton, "Driving recorder based on-road pedestrian tracking using visual SLAM and constrained multiple-kernel," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2014, pp. 2629–2635.
[25] C.-T. Chu, J.-N. Hwang, H.-I. Pai, and K.-M. Lan, "Tracking human under occlusion based on adaptive multiple kernels with projected gradients," *IEEE Trans. Multimedia*, vol. 5, no. 7, pp. 1602–1615, Nov. 2013.
[26] *"Gurobi Optimizer Reference Manual,"* Gurobi Optimization, Houston, TX, USA, 2012 [Online]. Available: http://www.gurobi.com
[27] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[28] P.-E. Forssén, "Maximally stable colour regions for recognition and matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.
[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[30] T. Ojala, M. Peitikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 1683–1698, Jul. 2002.
[31] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Cambridge, MA, USA: MIT Press, 2000.
[32] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[33] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 1201–1208.

Kuan-Hui Lee received the B.S. degree from the National Taiwan Ocean University, Keelung, Taiwan, in 2003, and the M.S. degree from the National Cheng Kung University, Tainan, Taiwan, in 2005, both in electrical engineering. He is currently working toward the Ph.D. degree in electrical engineering at the University of Washington, Seattle, WA, USA.

He was with the HTC Corporation, New Taipei City, Taiwan, from 2007 to 2009. His current research interests include computer vision, machine learning, and video/image processing.

Jenq-Neng Hwang (S'82–M'84–SM'96–F'01) received the B.S. and M.S. degrees, both in electrical engineering, from the National Taiwan University, Taipei, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree from the University of Southern California, Los Angeles, CA, USA, in 1988.

In 1989, he joined the Department of Electrical Engineering, University of Washington, Seattle, WA, USA, where he was promoted to Full Professor in 1999. He is currently the Associate Chair for Research with the Department of Electrical Engineering, University of Washington. He has authored or coauthored more than 300 journal papers, conference papers, and book chapters, including the textbook *Multimedia Networking: From Theory to Practice* (Cambridge University Press, 2009). His research interests include multimedia signal processing and multimedia networking.

Dr. Hwang is a founding member of the Multimedia Signal Processing Technical Committee, the IEEE Signal Processing Society, and was the society's representative to the IEEE Neural Network Council from 1996 to 2000. He is currently a member of the Multimedia Technical Committee of the IEEE Communication Society and also a member of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society. He served as Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and the *IEEE Signal Processing Magazine*. He is currently on the Editorial Board of ETRI, IJDMB, and JSPS. He was the Program Co-Chair of ICASSP 1998 and ISCAS 2009. He was the recipient of the 1995 IEEE Signal Processing Society's Best Journal Paper Award.