Adaptive Learning for Target Tracking and True Linking Discovering Across Multiple Non-Overlapping Cameras

Kuan-Wen Chen, Chih-Chuan Lai, Pei-Jyun Lee, Chu-Song Chen, *Member, IEEE*, and Yi-Ping Hung, *Member, IEEE*

***Abstract*—To track targets across networked cameras with disjoint views, one of the major problems is to learn the spatio-temporal relationship and the appearance relationship, where the appearance relationship is usually modeled as a brightness transfer function. Traditional methods learning the relationships by using either hand-labeled correspondence or batch-learning procedure are applicable when the environment remains unchanged. However, in many situations such as lighting changes, the environment varies seriously and hence traditional methods fail to work. In this paper, we propose an unsupervised method which learns adaptively and can be applied to long-term monitoring. Furthermore, we propose a method that can avoid weak links and discover the true valid links among the entry/exit zones of cameras from the correspondence. Experimental results demonstrate that our method outperforms existing methods in learning both the spatio-temporal and the appearance relationship, and can achieve high tracking accuracy in both indoor and outdoor environment.**

***Index Terms*—Brightness transfer function, camera network, non-overlapping cameras, spatio-temporal relationship, visual surveillance, visual tracking.**

I. Introduction

CAMERA networks are extensively used in visual surveillance because they can monitor the activities of targets over a large area. One of the main challenges of camera networks is to track targets across cameras or find the correspondence among cameras. Several studies [1], [5], [21], [34] have discussed multi-camera tracking with overlapping field of views (FOVs). However, it is difficult to always align the FOVs of cameras in practice. Compared to single camera tracking or tracking with overlapping FOVs, relatively little attention has been paid to target tracking across non-overlapping views. Difficulties for target tracking across disjoint views include 1) it is hard to calibrate the extrinsic parameters of cameras, 2) the target would disappear in all views for a while, and 3) illumination of distinct cameras could be highly different.

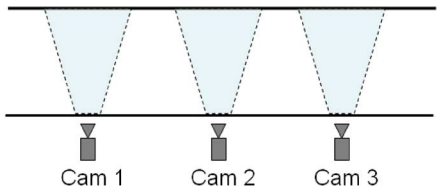

Fig. 1. Passage with three cameras and their FOVs.

Two visual cues employed for tracking targets across non-overlapping cameras are spatio-temporal cue and appearance cue. Although other cues could be used, such as relative size of target's bounding box [14] and target's moving speed [18], their robustness relies highly on camera pose, image quality, and image analysis error. The spatio-temporal relationship includes the transition time probability between two entry/exit zones belonging to different cameras. The appearance relationship describes the change of target's appearances taken by different cameras, and is often modeled as a brightness transfer function (BTF) that transfers the brightness distribution from one camera to another. Therefore, a key problem of tracking targets across multiple disjoint cameras is to learn both the spatio-temporal relationship and the brightness transfer function.

A. Characteristics of Our Approach

To learn the spatio-temporal relationships, we introduce a batch learning algorithm first, and extend it to update incrementally. In addition, a novel algorithm for removing weak links is proposed. Our method has two characteristics:

The first is to discover valid links (or remove weak links) for a camera network whose topology is initially unknown. The weak link is a link that does not exist in the real world, but it would be mistaken as a valid link. As shown in Fig. 1, there would be three valid links detected after learning from the traffic among cameras, i.e., Cam 1 to Cam 2, Cam 2 to Cam 3, and Cam 1 to Cam3. However, it is impossible for someone exiting from Cam 1 and entering Cam 3 without passing the view of

Manuscript received December 30, 2010; accepted March 04, 2011. Date of publication March 22, 2011; date of current version July 20, 2011. This work was supported in part by the National Science Council, Taiwan, under Grants NSC 98-2221-E-002-127-MY3 and NSC 98-2221-E-001-012-MY3, and in part by the Ministry of Economic Affairs, Taiwan, under Grant 99-EC-17-A-02-S1-032. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Paal Halvorsen.

K.-W. Chen is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

C.-C. Lai and P.-J. Lee are with the Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan.

C.-S. Chen is with the Institute of Information Science, and Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan and also with the Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan (e-mail: song@iis.sinica.edu.tw).

Y.-P. Hung is with the Institute of Networking and Multimedia and the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan and also with the Institute of Information Science, Academia Sinica, Taipei, Taiwan (e-mail: hung@csie.ntu.edu.tw).

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.2011.2131639

Cam 2, and so the link of Cam 1 to Cam 3 is a weak link. The following problems resulted from weak links become critical as the camera network becomes larger:

- *Increasing computational complexity*—When searching targets in a camera network, the weak links increase the searching space considerably.
- *Increasing tracking error*—The weak links cause much false correspondence when tracking targets in a camera network.
- *Confusing users when monitoring*—Some researches [15] have developed user interfaces based on the activity topology of cameras, so that a target can be followed across cameras more easily. When weak links exist, users will be confused as there might be multiple false candidates of camera views where the target may appear after a certain time interval.

To our knowledge, there are no previous works discussing the problem of weak links, and we provide a solution to remove them automatically.

The second characteristic is to learn the BTFs automatically and adaptively, without needs to providing the correspondence between cameras manually. We develop an unsupervised learning method that can be adaptive to illumination changes. The required training data is much less than that in Gilbert and Bowden [14]. Our method is thus adaptive to handle sudden illumination changes, a situation usually happens in the indoor environment and is unavoidable for long-term monitoring, but has not been handled by previous methods.

II. Related Work

There have been some notable works in tracking across non-overlapping cameras. Huang and Russell [17] first presented a probabilistic approach for object identification across two non-overlapping cameras. Then Pasula *et al.* [27] extended this approach for tracking through more than two sensors, and proposed a polynomial-time approximation algorithm based on Markov chain Monto Carlo simulation. Kettnaker and Zabih [20] employed a Bayesian formulation of the problem to reconstruct the paths of objects across multiple cameras, and transformed it into a linear-programming problem to establish correspondence. Porikli and Divakaran [28] proposed an inter-camera color calibration model to estimate the optimal alignment function between the appearance histograms of the objects in different views, and then combined spatio-temporal and appearance cues to track objects. Dick and Brooks [11] employed a stochastic transition matrix to describe the observed pattern of people motion. Prosser *et al.* [29] proposed a cumulative BTF for mapping colors between cameras. Javed *et al.* [18] presented a system that learned the camera network topology and path probabilities of objects using Parzen windows with manual correspondence in an initial training phase. They also proposed an appearance model, which showed that all brightness transfer functions from one camera to another lie in a low-dimensional subspace and learned the subspace for computing appearance similarity.

The methods mentioned above either assumed that the camera network topology and transition models are known, or fit them with hand-labeled correspondence or obvious markers. In practice, they could be difficult to implement under the real-world condition due to the complicated learning phase; in particular, when the environment changes (such as illumination changes), the above scenarios would fail to work.

Makris *et al.* [23] proposed a method which does not require hand-labeled correspondence. The method automatically validates a camera network model using within-camera tracking data. This approach has been extended by Stauffer [32] and Tieu *et al.* [36] by providing a more rigorous definition of a transition based on statistical significance. Stauffer [32] handled the nonstationary traffic processes resulted from traffic lights, vehicle grouping, and other nonlinear vehicle-to-vehicle interactions. The method of Tieu *et al.* [36] generalizes Makris *et al.*'s approach more flexibly with multi-model transition time distributions, and explicitly handles correspondence. Gilbert and Bowden [14] extended Makris *et al.*'s approach to incorporate coarse-to-fine topology estimations, and further proposed an incremental learning method to model the color variations, relative sizes, and posterior probability distributions of the spatio-temporal links between cameras.

To learn the spatio-temporal relationship, some of the above approaches [23], [32], [36] used batch learning procedures, and estimated the entry/exit zones in advance. However, there are some limitations with their methods. First, how much training data is required in the batch-learning procedure remains unclear. Second, if the environment changed, the only solution is to reboot the whole system. Gilbert and Bowden [14] learnt the spatio-temporal relationship incrementally, but the spatio-temporal links are block based instead of entry/exit zone based, while the later can usually be learnt from a single image efficiently. The number of blocks may grow quickly although a coarse-to-fine estimation method was proposed. The coarse-to-fine strategy may also limit the adaptability. For instance, if a camera is slightly moved or a new entry is opened, their method would fail to work.

There were some works [10], [30], [38] focusing on online camera topology learning, which is similar to our incremental learning approach. Shafique *et al.* [30] estimated the relative topology of overlapping cameras by employing the statistics of co-occurring observations. Detmold *et al.* [10] determined activity topology by using the exclusion algorithm, and Wang *et al.* [38] clustered trajectories in multiple camera views and learnt the common paths in the scene. These unsupervised methods could be applied to non-overlapping cameras without solving the correspondence problem. However, unlike the spatio-temporal relationships estimated in this paper, they found the camera topology only. That is, they only estimate the connectivities among cameras, but overlook the expected time (which usually reflects the relative distance) from one camera to another. In addition, the weak link problem discussed in Section I-A could still exist when directly applying these methods. In the experiments, we have demonstrated that the approach using the spatio-temporal relationships (*ST only*) can actually outperform the approach that uses camera topology only (the baseline method) for tracking, where the spatio-temporal relationships employ further the transition time information of each connected link.

To learn the BTF, to our best knowledge, Gilbert and Bowden method [14] is the only one that learns it without hand-labeled correspondence. It seems to be adaptive to illumination changes due to the incremental learning procedure. However, the method needs much learning data and learns without considering false correspondence; therefore, it works well when there is none or only slow illumination change. In the experiments, we have shown that much less training data are required by our method, and thus our method can be adaptive to even handling sudden illumination changes.

Opposite to matching the global color information, some works [12], [13], [19] employed local features or edge signatures for person re-identification. However, such approaches can only handle targets of similar poses observed with high quality.

III. Problem Formulation

Multi-camera tracking with disjoint views seeks to establish correspondence between observations of objects across cameras. This is often termed as object "handover," where one camera transfers a tracked object or person to another camera. The handover list is a set of observations having left from one camera view within the maximum allowable reappearance period T_{max}. Suppose that a person P enters the view of one camera and denote this observation as O_P, we could get from O_P the spatio-temporal cue $st(O_P)$ and the appearance cue $app(O_P)$. The $st(O_P)$ includes the information of the arrival camera id, location $s(O_P)$, and time $t(O_P)$.

Let O_h be an observation in the handover list. Denote the probability of the observation O_P belonging to O_h in the handover list as $p(P = h|O_P, O_h)$. The most likely correspondence could be obtained as follows:

$$h^* = \arg\max_{h \in H_s} (p(P = h|O_P, O_h)) \tag{1}$$

where H_s is the handover list. If the probability does not exceed a threshold, P is considered the new person arriving in the monitored environment. In the following section, for simplicity, we denote that the matching probability of two targets P and h as follows:

$$p_{match}(P, h) = p(P = h|O_P, O_h). \tag{2}$$

Assume equal priority, and the spatio-temporal cue and appearance cue are independent. We take log likelihoods and merge them by using a fusion weighting factor w. From Bayes Theorem, we have

$$\begin{aligned} h^* &= \arg\max_{h \in H_s} (\ln p(P = h|O_P, O_h)) \\ &= \arg\max_{h \in H_s} \Big(\ln \big(p\left(st(O_P), st(O_h)P = h\right)^w \\ &\qquad \times p\left(app(O_P), app(O_h)P = h\right)^{(1-w)}\big)\Big) \\ &= \arg\max_{h \in H_s} (w \times \ln p\left(st(O_P), st(O_h)P = h\right) + (1 - w) \\ &\qquad \times \ln p\left(app(O_P), app(O_h)P = h\right)). \end{aligned} \tag{3}$$

In (3), the term $p(app(O_P), app(O_h)|P = h))$ is the probability of appearance similarity between person P and h, which can be calculated as the histogram intersection or Bhattacharyya coefficient [6] after color-histogram transformation that will be detailed in Section VI.

The other term $p(st(O_P), st(O_h)|P = h))$ is the probability of spatio-temporal similarity. Suppose that the entry/exit zones of O_P and O_h are Z_P and Z_h, respectively, and $p_{ab}(t)$ is the transition time probability that someone takes a period of time t to transit from zone a to zone b. Then, we estimate $p(st(O_P), st(O_h)|P = h))$ by assuming that it is composed of two independent terms, the transition time probability (temporal part) and the zone location probability (spatial part):

$$\begin{aligned} &p\left(st(O_P), st(O_h)P = h\right) \\ &\quad = \sum_{Z_P} \sum_{Z_h} \underbrace{p\left(Z_P, Z_h, t(O_P), t(O_h)|P = h\right)}_{\text{temporal part}} \\ &\qquad \times \underbrace{p\left(s(O_P)|Z_P, P = h\right) p\left(s(O_h)|Z_h, P = h\right)}_{\text{spatial part}} \\ &\quad = \sum_{Z_P} \sum_{Z_h} p_{Z_P Z_h}(t) p\left(s(O_P)|Z_P\right) p\left(s(O_h)|Z_h\right) \end{aligned} \tag{4}$$

where $p_{Z_P Z_h}(t)$ is the transition time probability distribution with $t = t(O_P) - t(O_h)$, and $p(s(O_\bullet)|Z_\bullet)$ is the zone location probability of the observation $O_\bullet$ entering or exiting from the zone $Z_\bullet$, which is a Gaussian mixture model (GMM) learnt for the entry/exit zones as will be introduced in the following. The probability of spatial temporal similarity will then be detailed in Section IV.

IV. Learning Spatio-Temporal Relationship

In our method, the spatio-temporal relationships are entry/exit zone based [23], [32], [36], and the learning procedures contain two phases: batch learning phase and incremental learning phase. What we want to learn are the entry/exit zones for each camera, the transition time probability distribution between each pair of zones, and which pairs of zones are connected, called valid links.

A. Batch Learning Phase

In the batch learning phase, we estimate the entry/exit zones for each single image at first. We gather entry/exit points for each camera view with the results of single camera tracking. In each camera view, we model the entry/exit zones as a GMM and use expectation maximization (EM) algorithm to estimate the parameters of GMM [9], [22]. The number of clusters is determined automatically according to Bayesian information criterion (BIC).

After the estimations of entry/exit zones, we create possible links for all pairs of zones belonging to different cameras. Then, the transition time probability distribution is learnt for each possible link. Here, our approach is similar to that of [14], except that it is entry/exit zone based instead of block based, where the later is much more inefficient and difficult to be extended as adaptive to environments. Suppose that there is a possible link between two entry/exit zones, zone a is in the view of camera 1 and zone b is in the view of camera 2. Denote $p_{ab}(t)$ to be the transition time probability that someone takes a period of time t to move from zone a to zone b, and T_{max} to be the maximum

allowable reappearance period. The object i exits from the zone a at the time t_i. The object j enters the zone b at the time t_j. S_{ij} is the appearance similarity between the objects i and j with the detail being defined in Section VI. Then, the transition time probability distribution is calculated as

$$p_{ab}(t) = \frac{1}{C}\sum_i \sum_j \begin{cases} S_{ij}, & \text{if } (t_j - t_i) = t \\ 0, & \text{otherwise} \end{cases} \quad t < T_{\max}, t \leq 0 \tag{5}$$

where $C = \sum_{t=0}^{T_{\max}} p_{ab}(t)$ is a normalization term.

After estimating the transition time probability distribution, we measure the noise floor level for each link. The noise floor level T_{nfl} in our system is set as the double of the median value. If sufficient evidence T_{se} (set as 20 in our implementation) has been accumulated and the maximum peak of the distribution exceeds the noise floor level, the possible link is set initially as a valid link between zones:

$$\begin{cases} Min\,(Count(i), Count(j)) > T_{se} \\ Max\,(p_{ab}(t)) > T_{nfl}\,(\text{where } T_{nfl} = 2 \times Median\,(p_{ab}(t))) \end{cases} \\ \Rightarrow \text{The link } L_{ab} \text{ is set as valid.} \tag{6}$$

The rational of (6) is explained as follows. Since the time for a person moving from one zone to another is not randomly distributed, most work assumes implicitly that the true correspondence will lead to a highly structured distribution, and we suppose that the variance is σ^2. A small variance σ^2 enables the true transition time distributing over fewer bins and makes the maximum peak of the distribution exceed the noise floor level more easily. This is the reason why (6) can be used to detect the initial candidates of valid links. However, as we have described in Section I, the weak links could be mistaken as valid links. In Section V, we propose an algorithm to find the weak links automatically.

B. Incremental Learning Phase

A main characteristic of our approach is the incremental adaptation of both spatio-temporal and appearance information, so that our approach can be adaptive to environment changes. In the following, we focus on the incremental adaptation of spatio-temporal information, and the adaptation of appearance similarity measurements will be introduced in Section VI. In the incremental learning phase, we update the entry/exit zones and transition time probability with time. According to the definition of transition time probability between zones as shown in (5), learning it incrementally is simple. It updates with each time occurrence of someone entering the FOV of a camera. Suppose that someone P enters the FOV of a camera, then we seek all possible candidates $H = \{h_1, \ldots, h_k\}$ in the handover list, satisfying that $t(P) - t(H) \leq T_{max}$. Then, we update all possible links of each pair of Z_P and Z_H by (5), where Z_P and Z_H are the most possible entry/exit zones of P and H, respectively, according to the probabilities of GMM.

There are some problems with the batch learnt entry/exit zones [23], [32], [36]. First, it is likely to misclassify two zones into one single zone when the two zones are adjacent in the image, as shown in Fig. 2. On the other hand, it could possibly divide a zone into several smaller zones. Second, the environment may change due to camera addition/removal. We may also lack the training data when there are no objects entering a room or passing some passages during the data-collection period of the batch learning phase. For solving the problem, we update the entry/exit zones by using the online K-means approximation principle [33] to update the Gaussian mixture model with each time occurrence of someone entering the FOV of a camera as well. In [33], the online K-means approximation is a renowned method used for background modeling, where the Gaussian models in the 3-D RGB color space are used. In this work, we employ the same principle to the incremental learning of the entry/exit zones, where the Gaussian models in a 2-D x-y space are used as illustrated in Fig. 2. Suppose that someone P enters the FOV of a camera, $H = \{h_1, \ldots, h_k\}$ are all possible candidates in the handover list. The matched zones Z_p and Z_h are selected by the following equation:

$$(Z_P, Z_h) = \underset{Z_P, Z_h | h \in H}{\arg\max} \left(p_{Z_P Z_h}(t) p\left(s(O_P)|Z_P\right) p\left(s(O_h)|Z_h\right)\right) \tag{7}$$

where $s(O_\bullet)$ [shown in Fig. 2(a)] is the coordinate of object $O_\bullet$ and $p(s(O_\bullet)|Z_\bullet)$ is the probability that $O_\bullet$ occurs at this zone. Then the matched zones are updated with online K-means approximation [33]. Assume the mean and covariance matrix of a matched zone are $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and the new observation is $\mathbf{X}_t$, where $\mathbf{X}_t \in \mathrm{R}^2$. The parameters of the distribution are updated as follows:

$$\begin{cases} \boldsymbol{\mu}_t = (1-\rho)\boldsymbol{\mu}_{t-1} + \rho \mathbf{X}_t \\ \boldsymbol{\Sigma}_t = (1-\rho)\boldsymbol{\Sigma}_{t-1} + \rho(\mathbf{X}_t - \boldsymbol{\mu}_t)^T(\mathbf{X}_t - \boldsymbol{\mu}_t) \end{cases} \tag{8}$$

where ρ is the learning factor for adapting current distribution. Notice that we update zones by using not only the location information $p(s(O_\bullet)|Z_\bullet)$ but also the transition time information in (7), and it improves the estimation of zones incrementally.

In addition, as the number of Gaussian models in GMM is hard to determine in advance and could also be varying with time, we propose two operators for learning zones incrementally: *Zone Merging* and *Zone Split*. The *Zone Merging* merges two Gaussian models in GMM [31]. In [31], Song *et al.* proposed a clustering merging strategy to merge Gaussian models for online data stream clustering. In our implementation, the *Zone Merging* is applied when the distance of mean of two zones are near enough (set as 32 pixels in our implementation) and found to have similar distributions and valid links to other zones. After each zone updating, we use Hotelling's T-*square test* and W *statistic test* for testing the mean equality and covariance equality, respectively, and then determine whether two

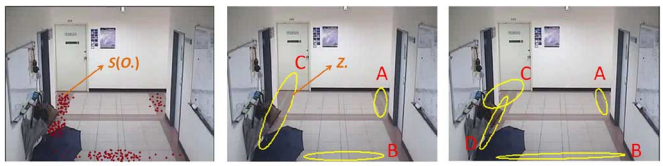

Fig. 2. Example of entry/exit zone estimation for E2_Cam 3 in Fig. 5. (a) Gathered entry/exit points. (b) Entry/exit zones, A, B, and C, after batch learning phase. (c) Entry/exit zones, A, B, C, and D, after incremental learning phase. Note that zones C and D are very close. The zone C in (b) is split and adapted into zones C and D in (c) incrementally.

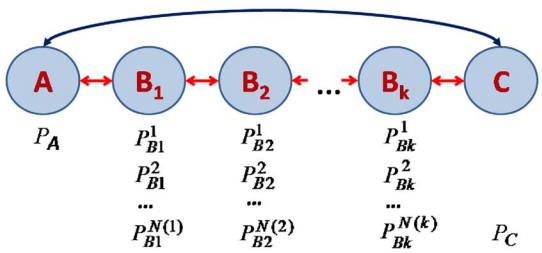

Fig. 3. Node A, C, and $B_\bullet = \{B_1, B_2, \ldots, B_k\}$ are entry/exit zones. The red lines are true valid links, and the blue line is a weak link. $P_\bullet$ are all targets within the time interval of $t(P_A)$ and $t(P_C)$, where $N(z)$ is the number of $\{P^n_{B_z}\}$ in B_z.

zones are needed to be merged. Two zones are merged by the following equations:

$$\begin{cases} \boldsymbol{\mu} = \frac{w_i\boldsymbol{\mu}_i + w_j\boldsymbol{\mu}_j}{w_i+w_j} \\ \boldsymbol{\Sigma} = \frac{w_i\boldsymbol{\Sigma}_i + w_j\boldsymbol{\Sigma}_j}{w_i+w_j} + \frac{w_i\boldsymbol{\mu}_i\boldsymbol{\mu}_i^T + w_j\boldsymbol{\mu}_j\boldsymbol{\mu}_j^T}{w_i+w_j} - \boldsymbol{\mu}\boldsymbol{\mu}^T \\ w = w_i + w_j \end{cases} \tag{9}$$

where $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ are the parameters of two merged zones, and w_i and w_j are their weights in GMM, respectively. The $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}$, and w are the parameters of the new zone after merging.

The *Zone Split* is used for solving the problem of misclassifying two zones into one single zone as shown in Fig. 2. It is applied for a zone whose Gaussian model has a large variance value (set as 32 pixels in our implementation), and has two valid links to different zones. A big zone can be separated into two isolated zones when the zone links to two or more locations in the environment. In most situations, the leaving position of a target is usually related to where it will be. For instance, a person leaves from the upper portion of the zone C in Fig. 2(b) would more probably go to the upper place, i.e., the E2_Cam 1 in Fig. 5(b). For the zone to be split, we generate a new Gaussian model with almost the same settings as the original one, except the valid links. We assign these two zones with different valid links and halve their mixture weights by assuming that they have equal weights. Due to the different transition time probability distributions, two Gaussians will be evolved to isolated zones according to (7).

V. Automatic Discovering and Removing Weak Links

In our algorithm, a link is initially considered as a valid link if the maximum peak of the transition-time distribution exceeds the noise floor level. Unfortunately, some of the valid links are the weak links (as mentioned in Section I-A), because there is also correlation between the departure and arrival times belonging to the weak links. As shown in Fig. 3, suppose nodes A, C, and $B_\bullet = \{B_1, B_2, \ldots, B_k\}$ are entry/exit zones belonging to different cameras, and there are valid links $\{AB_1, B_1B_2, B_2B_3, \ldots, B_{k-1}B_k, B_kC\}$ and one weak link L_{AC} between A and C. For simplicity, we consider a typical case where a target passing A, $B_\bullet$, and C in turn. For each target P_C entering C, P_A is the most possible corresponding target in A. $\{P^n_{B_z}\}$ are all targets in B_z that $t(P^n_{B_z})$ is within the time window between $t(P_A)$ and $t(P_C)$, i.e., $t(P_A) < t(P^n_{B_z}) < t(P_C)$, for all z and n, where n is the index of targets in B_z. The transition of the weak link L_{AC} is resulted from the transitions of passing the valid links from A to $B_\bullet$ to C. From the characteristics of weak links, we give the following two propositions that are useful in removing weak links. First, the average transition time of a weak link is larger than the sum of the average transition times of the corresponding valid links. This can be formally stated as follows.

Proposition 1: Suppose a weak link L_{AC} is between node A and C, and there are corresponding valid links between node A and B_1, B_1 and $B_2, \ldots, B_{k-1}$ and B_k, and B_k and C. Then we have

$$E(t_{AC}) > E\left(t_{AB_1}\right) + \sum_{n=1}^{k-1} E\left(t_{B_nB_{n+1}}\right) + E\left(t_{B_kC}\right) \tag{10}$$

where $E(t_{ij})$ is the expected value of transition time between nodes i and j.

Proposition 1 is held simply because the passing time through A to C shall contain further the passing times spent for every single node, $B_1, B_2, \ldots, B_k$.

Second, if a target P_A in node A and a target P_C in node C are the same, then there exists a target P_B in the nodes of $B_\bullet$, which is the same as P_A and P_C.

Proposition 2: Consider the same condition of **Proposition 1**. Let $P \in B$ denote that the target P is an observation of node B, and $SAME(P_i, P_j)$ means P_i and P_j are the same target. For every corresponding observations $P_A \in A$ and $P_C \in C$, i.e., $SAME(P_A, P_C)$, then

$$\exists P_{Bz} \in B_z \; s.t. \; SAME(P_{Bz}, P_A), \quad for \; z = 1 \; to \; k. \tag{11}$$

Proposition 2 is easily held since the target shall appear in every single node. Note that there could be occlusions in some nodes where the target disappears. To handle this problem, we assume that the target could be occluded only in a few nodes. We use dynamic programming to find a global-matching cost when employing **Proposition 2** for weak link removal as introduced below, so that occlusions that happen in a limited number of nodes can be tolerated.

A. Remove Weak Links—Batch Learning Phase

From the above propositions, we first introduce a batch learning algorithm for removing the weak links in Section V-A. Then we present an incremental learning algorithm that can avoid building weak links in Section V-B. After learning the spatio-temporal relationship, the entry/exit zones can be viewed as nodes in an undirected graph G with both true valid links and weak links. The length of each link is assigned by the expected value of transition time between two entry/exit zones. In the batch learning phase, we first apply **Proposition 1** to find a candidate weak link L_{AC} which satisfies the following conditions: 1) it is the longest link, 2) it belongs to a cycle, and 3) its length is larger than the sum of the length of the other links in the cycle. To search candidate weak link L_{AC}, the longest link can be found with time-complexity $O(n_l)$, where n_l is the number of links in the graph. To check whether the candidate link L_{AC} belonging to a cycle has a length larger than the sum of the lengths of the other links in the cycle, we remove link L_{AC} and resulted in a subgraph G'. Then we find

the shortest path from node A to C in G' and check whether the length of L_{AC} is larger than that of the shortest path. Its time-complexity is $O(n_l + n_v \log n_v)$ by Dijkstra's algorithm, where n_v is the number of nodes in the graph.

We then employ **Proposition 2** for further verification of the valid-link candidates obtained above. We find a target P_A with maximal probability of $p_{\text{match}}(P_C, P_A)$ defined in (2) for each entering event P_C in C. If $p_{\text{match}}(P_C, P_A)$ is larger than a threshold, we choose a set of target $\{P_{Bz}^n\}$ in B_z such that $t(P_{Bz}^n)$ is within the time interval between $t(P_A)$ and $t(P_C)$. Then, by **Proposition 2**, we consider a subgraph connected by P_A, $\{P_{Bz}^n\}$, and P_C only, and assign the weights of its links by using the matching probabilities. To check whether the condition of **Proposition 2** holds, we apply dynamic programming (DP) [7] for finding a path from P_A to P_C with the maximal probability p_{ABC}:

$$p_{ABC} = \max_{P_{B_1}, P_{B_2}, \ldots, P_{Bk} \in \{P_{Bz}^n\}} (p_{match}(P_A, P_{B_1}) \times p_{match}(P_{B_1}, P_{B_2}) \times \cdots \times p_{match}(P_{B_k}, P_C)). \quad (12)$$

From **Proposition 2**, if L_{AC} is a weak link, there must be such a path of high probability.

After applying both Propositions 1 and 2, we obtain a set of candidates that could be weak links. Considering a candidate weak link L_{AC}, we use the matched targets between nodes A and C to decide whether or not it is weak. For each matching pair of targets (P_A, P_C), we vote for the link L_{AC} a weak link with the weight p_{ABC}, and record the accumulation score in the variable S_{Weak} if p_{ABC} is larger than a threshold. On the other hand, if L_{AC} is not a weak link, p_{ABC} would be small or close to zero. We then vote for the L_{AC} a valid link with the weight $p_{\text{match}}(P_C, P_A)$ to the accumulation score S_{Valid}. After voting for each matching pair of P_A and P_C, if the score S_{Weak} is much larger than S_{Valid}, this link is deemed as a weak link and removed. Then we repeat the steps until there are no possible weak links. Details are shown in Algorithm 1.

Algorithm 1. Algorithm for removing weak links

1. **loop for each valid link** $S_{Weak} = 0$, and $S_{Valid} = 0$.
2. According to Proposition 1, find a candidate link L_{AC} satisfying
3. 1) $E(t_{AC})$ is maximal among unchecked links.
4. 2) Belong to a cycle.
5. 3) $E(t_{AC}) > E(t_{AB_1} + \sum_{n=1}^{k-1} E(t_{B_n B_{n+1}}) + E(t_{B_k C})$
6. **loop for each arrival** $\boldsymbol{P_C}$ (or $\boldsymbol{P_A}$)
7. **if** $\exists P_A$ (or P_C) s.t. $(P_{match}(P_C, P_A) > T_1)$ **then**
8. From **Proposition 2**,
9. 1) Establish a weighted graph with nodes P_A, $\{P_{Bz}^n\}$, and P_C and assign weight of each link by the matching probability.
10. 2) Solve a path from P_A to P_C (or from P_C to P_A) with maximal probability p_{ABC} by DP.
11. 3) **if** $p_{ABC} > T_2$ **then**
12. $S_{Weak} = S_{Weak} + p_{ABC}$.
13. **else**
14. $S_{Valid} = S_{Valid} + p_{\text{match}}(P_C, P_A)$.
15. **end if**
16. **end if**
17. **end loop**
18. **if** $S_{Weak} > (T_3 \times S_{Valid})$
19. Remove weak link L_{AC}
20. **end if**
21. **end loop**

B. Remove Weak Links—Incremental Learning Phase

Although the computation of the batch learning algorithm introduced above is practically affordable, we use the batch algorithm for only a short period of time and rely mostly on the incremental learning introduced below to avoid weak links. Hence, the computation required can be further reduced. In the batch learning, a graph containing both valid and weak links has already been built, and then we seek to remove the weak links. This is much different to our incremental learning by which we avoid producing weak links and build the valid-link graph directly. Our incremental learning algorithm mainly relies on the **Proposition 3** below, which shows that if L_{AC} is a weak link, it will be learnt later than all of the corresponding true valid links $L_{AB_1}, L_{B_1B_2}, \ldots, L_{B_{k-1}B_k}, L_{B_kC}$ by using (6).

Unlike Propositions 1 and 2 that hold generally, **Proposition 3** holds with some conditions often satisfied in practice. Denote the transition time from A to C to be a random variable $Y = |t(P_C) - t(P_A)|$ with variance σ_{AC}^2, and that of the internal links L_{ij} to be a random variable $X_{ij} = |t(P_i) - t(P_j)|$. To simplify the notations, A and C are denoted as B_0 and B_{k+1} below, respectively. Then we have

$$Y = \sum_{i=0}^{k} X_{(i)(i+1)} + \sum_{i=1}^{k} X_{Bi} \quad (13)$$

where X_{Bi} is the staying time of target at node B_i for $i = 1$ to k. Considering one of the true valid links from nodes B_r to B_{r+1}

$$Y = X_{(r)(r+1)} + \sum_{\forall i | (i \neq r)} X_{(i)(i+1)} + \sum_{i=1}^{k} X_{Bi} = X_{(r)(r+1)} + X'_{(r)} \quad (14)$$

where

$$X'_{(r)} = \sum_{\forall i | (i \neq r)} X_{(i)(i+1)} + \sum_{i=1}^{k} X_{Bi} \quad (15)$$

is the sum of the other random variables and is also a random variable with variance $\sigma_{X'}^2$. Hence, we have

$$\sigma_{AC}^2 = \sigma_{(r)(r+1)}^2 + \sigma_{X'}^2 + 2 \times Cov\left(X_{(r)(r+1)}, X'_{(r)}\right) \quad (16)$$

where C_{ov} stands for the covariance between two random variables.

Proposition 3: Consider the same condition of **Proposition 1**. As long as the covariance $Cov(X_{(r)(r+1)}, X'_{(r)})$ is larger than or equal to zero for all r, the weak link L_{AC} will be learnt later than all of the corresponding true valid links $L_{AB1}, L_{B1B2}, \ldots, L_{Bk-1Bk}$, and L_{BkC} by using the decision criterion (6).

To explain **Proposition 3**, let us re-investigate the rational of (6). Remember that a smaller variance enables the true transition time distributing over fewer bins and makes the maximum peak of the distribution exceed the noise floor level more easily, and thus (6) can be used to detect the initial valid links. According to (16), since $Cov(X_{(r)(r+1)}, X'_{(r)}) \geq 0$, we have

$$\sigma^2_{AC} \geq \sigma^2_{(r)(r+1)}. \tag{17}$$

That is, the variance of transition time distribution of a weak link is larger than that of every corresponding true valid link. Hence, weak links will be learnt later than all of the corresponding true valid links via (6).

In the following, we explain why the condition $Cov(X_{(r)(r+1)}, X'_{(r)}) \geq 0$ holds quite often in practice. Because $X_{(r)(r+1)}$ and $X'_{(r)}$ represent the transition times of a target moving across different locations, the covariance depends on the variation of velocity during the transition period. When the covariance of $X_{(r)(r+1)}$ and $X'_{(r)}$ is positive, the velocities of a target passing through $L_{(r)(r+1)}$ and the other links are positively correlated. That is, when considering the mean speed $mean\_spd_{(r)(r+1)}$ of all the targets passing through $L_{(r)(r+1)}$ and the mean speed $mean\_spd'_{(r)}$ of all the targets passing through the other links, we assume that most people who walk from B_r and B_{r+1} faster than the mean speed $mean\_spd_{(r)(r+1)}$ will pass through the other links faster than the mean speed $mean\_spd'_{(r)}$, too. Similarly, most people who walk slower than mean from B_r and B_{r+1} will pass through the other links slower than the mean. This is the most common traffic condition. The case of the covariance of $X_{(r)(r+1)}$ and $X'_{(r)}$ being negative is unusual. It happens when sufficiently many targets pass through $L_{(r)(r+1)}$ faster than the average speed of persons $(mean\_spd_{(r)(r+1)})$ but pass across the other links slower than the average speed of persons $(mean\_spd'_{(r)})$; simultaneously, there should also be sufficient many targets move slower than $mean\_spd_{(r)(r+1)}$ for $L_{(r)(r+1)}$ but faster than $mean\_spd'_{(r)}$ for the other links, or the means cannot be maintained as the assumed values. This situation could happen but is rare in general. Furthermore, even when the targets just walk around randomly (e.g., he/she may see around in the scene), the covariance of $X_{(r)(r+1)}$ and $X'_{(r)}$ is zero that still satisfies our assumption. The above examples thus explain why the assumption holds very often in practical situations.

According to **Proposition 3**, if we verify whether a new link is weak before creating it, then the links which have been created must be true valid links. Verifying each new link is similar to that of the batch learning algorithm, but we need to test only one candidate link (i.e., the newly formed link) without needing to test the other links. If the newly formed link satisfying the conditions of candidate weak link (in lines 3–5 of Algorithm 1), it will be verified before creating (by using lines 6–20). Otherwise, it is created directly. The newly added link will be considered a true valid link or a weak link after sufficient evidence (set as 50 in our implementation) has been accumulated.

VI. Learning Brightness Transfer Function

In Section III, the term $p(app(O_P), app(O_h)|P = h))$ is the probability of appearance similarity between person P and h after appearance transformation by the BTF. The appearance $app(O)$ is modeled as a normalized histogram, because it is relatively robust to changes in object pose [35]. In this section, we introduce an automatic method for learning a low-dimensional subspace of BTFs [18]. Notice that although the model of BTF is the same as that used by Javed *et al.* [18], their work learnt it with hand-labeled correspondence. On the contrary, we have proposed a new framework to learn it automatically [4].

A. Brightness Transfer Functions—A Review

Let f_{ij} be the BTF for every pair of observations O_i and O_j in the training set, and denote F_{ij} the collection of all the brightness transfer functions. Assume that the percentage of image points in O_i with brightness less than or equal to B_i is equal to the percentage of image points in O_j with brightness less than or equal to B_j. If H_i and H_j are normalized cumulative histograms of O_i and O_j, respectively, we can obtain the BTF f_{ij} as follows:

$$f_{ij}(B_i) = H_j^{-1}\left(H_i(B_i)\right) \tag{18}$$

where H^{-1} is the inverted cumulative histogram. Javed *et al.* [18] have presented a celebrated property: Giving a set of corresponding pairs of correct matches, their BTFs obtained via (18) will lie in a low-dimensional subspace. For learning the low-dimensional subspace, they use the probabilistic principal component analysis (PPCA) [18], [37]. Then, a d-dimensional BTF, f_{ij}, can be written as

$$f_{ij} = Wy + \overline{f_{ij}} + \omega \tag{19}$$

where y is a normally distributed q-dimensional subspace variable, $q < d$, and W is a $d \times q$ dimensional projection matrix. $\overline{f_{ij}}$ is the mean of collection of BTFs, and ω is isotropic Gaussian noise, i.e., $\omega \sim N(0, \sigma^2 I)$. Given that y and ω are normally distributed, the distribution of f_{ij} is

$$f_{ij} = N(\overline{f_{ij}}, Z) \tag{20}$$

where $Z = WW^T + \sigma^2 I$. More details can be found in [18] and [37].

We also verify this property by giving an example below. Considering the link between E2_Cam 1 and E2_Cam 3 in Fig. 5(b), we label 136 pairs of correspondence manually and then divide them into two classes: 36 pairs for training and 100 pairs for testing. A correct subspace of BTFs is learnt from the 36 pairs of correct correspondence. An incorrect subspace of BTFs is leant from 36 pairs of incorrect correspondence, which are produced by matching the correspondence randomly. The dimension of the subspace is fixed as 15 in both cases. Then, we estimate the reconstruction error distribution by testing the data that comes from the 100 pairs of correspondence

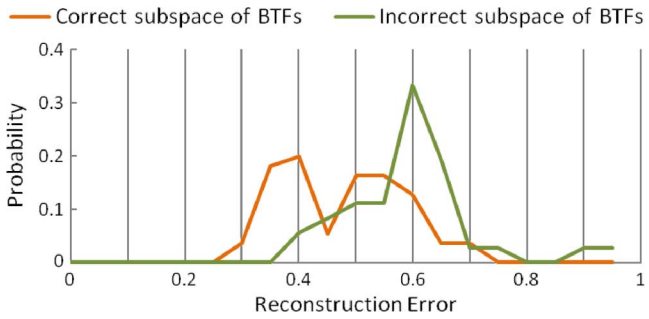

Fig. 4. Example of the reconstruction error distribution estimated by testing the hand-labeled correspondence with 50% matching accuracy.

with 50% matching accuracy. As shown in Fig. 4, a correct subspace of BTFs would have a more diverse reconstruction error distribution and lower errors than the one learnt by using incorrect correspondence.

B. Criterion for BTF Estimation

To estimate the BTF, the most critical part is to find the correspondence between two views. While Javed *et al.* [18] provided the correspondence manually, we find the correspondence automatically. The main principle we employed is that the BTFs lie in a low-dimensional space that can be represented by PPCA as observed in [18]. As shown above, when given correct correspondence, a correct low-dimensional subspace of BTFs can be learnt. Then, if a new pair of observations O_i and O_j belong to the same object, the reconstruction error of O_i and O_j would be small. On the contrary, if the observations O_i and O_j belong to different objects, the reconstruction error would be large.

Given n pairs of samples N, we can learn the PPCA subspace based on s pairs of subsamples $S(S \subseteq N)$, and obtain the reconstruction error of each pair. We then propose a criterion $p(\pi)$ for BTF estimation, where π is the sampled learning data set for BTF. We denote $similarity(pair_i)$ to be the similarity score of the ith corresponding pair, which is calculated by $(1 - reconstruction\_error(pair_i))$. Then, the criterion is

$$p(\pi) = \underset{i \in N-S | similarity\ (pair_i) > T_c}{mean} (similarity(pair_i)) \tag{21}$$

where T_c is a threshold decided by Otsu's thresholding algorithm [26].

C. Spatio-Temporal Information and MCMC Sampling

Our method finds the correspondence by constructing the low-dimensional subspace of the BTFs. If the correspondence is correct, the learnt subspace is expected to represent the BTFs well, and the reconstruction errors shall be smaller. The subspace of BTFs with fixed dimension (set as 15 in our implementation) is thus learnt without hand-labeled correspondence by sampling the training data set and choosing the best subspace of BTFs according to the criterion described in Section VI-B. However, it is not practical to sample all of the permutations directly. If there are n observations in both cameras, the number of matching permutations are $n!$, but the correct correspondences are at most n pairs. To solve this problem by more efficient searching, we use the spatio-temporal information and Metropolis-Hastings algorithm [16] for Markov chain Monte Carlo (MCMC) sampling.

We get n pairs of target correspondence N and their corresponding probability by using the spatio-temporal relationship. According to the experiments, the ratio of correct match is more than 60% by using the spatio-temporal cue only, which means that more than half of the n pairs are correctly matched. Then, we can sample s pairs $S(S \subseteq N)$, denoted by π, for learning the subspace of BTFs, where $s(s \ll n)$ is the number of corresponding pairs needed for learning. By sampling R times and s pairs per time, we choose the best one based on the criterion, (21), to test the remainder data $N - S$. We sample the learning pairs by using MCMC [8], [36] and use Metropolis-Hastings algorithm (Algorithm 2) [16], [36]. To our knowledge, the MCMC-sampling of correspondence was used by Tieu *et al.* [36] in estimating transition delay distributions (the temporal relationship between cameras) for determining the statistical dependence between two cameras, but it has not been used for finding the appearance relationship before.

Alogirthm 2. Metropolis-Hastings algorithm

1. Initialize π_0; $r = 0$.
2. **loop**
3. Sample π' from $q(\cdot|\pi)$.
4. Sample U from $U(0, 1)$.
5. Let $\alpha(\pi_r, \pi') = \min(1, (p(\pi')q(\pi_r|\pi')/p(\pi_r)q(\pi'|\pi_r)))$
6. **if** $U \leq \alpha(\pi_r, \pi')$ **then**
7. $\pi_{r+1} = \pi'$.
8. **else**
9. $\pi_{r+1} = \pi_j$.
10. **end if**
11. $r = r + 1$.
12. **end loop**

In detail, the initial sample π_0 is based on the corresponding probability. New samples π' are obtained given the current one π_r via a proposal distribution $q(\pi'|\pi_r)$, where π_r is the rth rounds of sampling results. We employ four types of proposals for $q(\pi'|\pi_r)$. First, *swap a pair*, and this swaps one of the s pairs chosen in the last time for one of the other $(n - s)$ pairs. Second, *jump*, and this re-samples the whole s pairs. Third, *add a pair*, i.e., incrementing s by 1. Fourth, *subtract a pair*, i.e., subtracting s by 1. The third and fourth proposals are used for avoiding s being decided incorrectly. The new sample is accepted with a probability proportional to the relative likelihood of the new sample versus the current one. The likelihood is proportional to the criterion (21). After executing the algorithm, the best sampling result recorded is chosen for learning the subspace of BTFs.

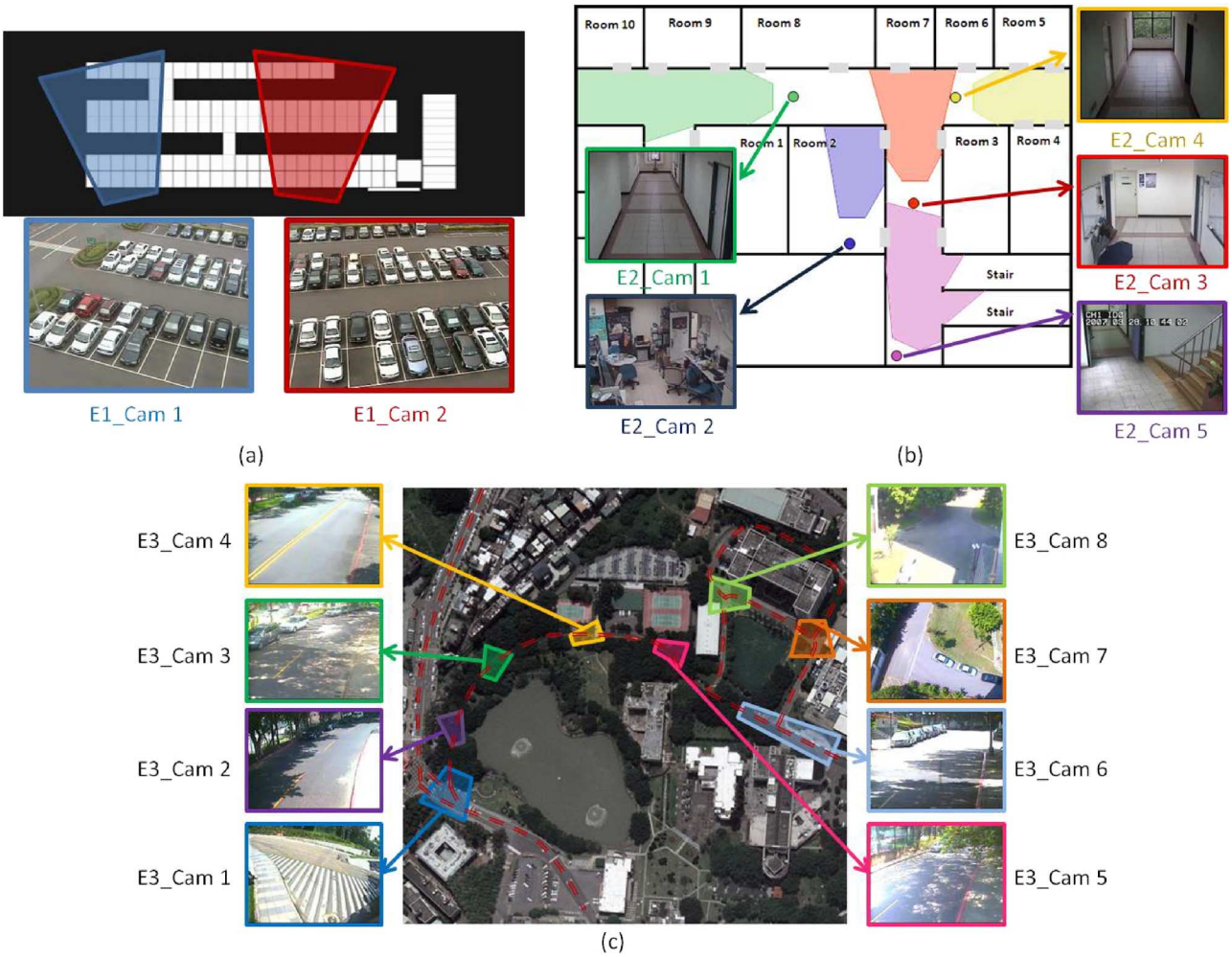

Fig. 5. Three experimental environments. (a) Parking lot. (b) Building. (c) Campus.

D. Adaptively Learning BTF

To make the BTF adaptive to illumination changes of the environment, we update the subspace of BTFs adaptively. When gathering a constant period of data, we learn the low-dimensional subspace of BTFs by the method introduced in Sections VI-A–VI-C, and gather the correct matching data, which is used for learning the subspace of BTF. We then update the old PPCA with the new arrival data by incremental PPCA [25]. In Fig. 9, it shows that the reconstruction error decreases with more data collected. Furthermore, it is adaptive to gradual illumination changes.

E. Handling Sudden Illumination Change

The weighting factor w, described in (3), indicates the confidence of learnt spatio-temporal relationship and BTF. It is adaptive to the changes of environments and can be used for handling sudden illumination changes. In our implementation, we employ the method introduced in [39] for handling sudden lighting changes. Once the sudden illumination change is detected, we set the weighting factor w to a higher value, which is set as 0.95 in our approach, i.e., the spatio-temporal cue is more reliable than the appearance cue. The BTF is initialized as an identity matrix. The weighting factor will decrease to a stable value, which is set as 0.6 and 0.4 for outdoor and indoor environments, respectively, in our approach, with respect to our adaptive learning procedure. The weighting factor could also be determined by the method in [3], which learns the weighting factor unsupervisedly and can be applied even when the testing environments are unknown. Note that our method learns a BTF by using a few data only, and so it adapts to illumination changes soon.

VII. Experimental Results

In this section, we evaluate the performance of our learning algorithm in three environments with no prior knowledge of camera network topology given in advance [4]. We compare our method with the state-of-the-art methods first, and demonstrate the superiority of our method. Then, four approaches are used to show that both relationships will improve the performance of tracking. Finally, we discuss the results and some difficulties encountered in real applications.

A. Experimental Setup

We perform the single camera tracking by using the method [33]. The same tracking results of single camera are used for all the experiments. We record the following information for each target in each camera view: entering time, entering location, exiting time, exiting location, and appearance histogram. The ap-

pearance histogram is extracted from the foreground pixels and 256-bin is used for each RGB color channels.

The implementation of our learning algorithm consists of a batch learning phase and an incremental learning phase. At bootup, the system collects observation data for a period of time. Then, the entry/exit zones of each camera view are learnt first. Next, we learn the transition time probability of possible links. After that, we apply the proposed method for removing weak links. Finally, we learn the low-dimensional subspace of BTFs for each pair of cameras, which depends on the spatio-temporal relationships.

When tracking targets across multiple cameras as described in Section III, the system simultaneously updates both the spatio-temporal relationships and appearance relationships. The transition time probability distribution and entry/exit zones are updated for each arrival event. The subspaces of BTFs are updated after collecting matching data of a predefined interval.

We evaluate our method in three different environments. The first experimental environment is shown in Fig. 5(a). It is an outdoor parking lot and contains two cameras. We record a 2-h period during the rush hour in the morning, and set T_{max} and ΔT, where ΔT is the quantized time interval of the transition time probability distribution, as 30 and 3 s, respectively. There is a property deserving notice in this environment: the transition time is varied in different situations. When the parking lot is empty or nearly empty, the transition time between cameras is small, about 7.6 s. However, if the parking lot is almost full, the transition time between cameras will become large, about 15.1 s. This is because the purpose of vehicles passing in this environment is to find an empty stall. In our testing sequences, the parking lot is nearly empty in the beginning, and it becomes almost full after 30 min.

The second environment is shown in Fig. 5(b). It is an indoor environment containing five cameras. We record a 6-h period in the daytime and set T_{max} and ΔT as 15 and 2 s, respectively. The main properties of this environment are 1) unlike the outdoor environment that the major targets tracked are vehicles which usually keep the distance, the persons tracked in an indoor environment are likely to walk together, and 2) the illumination condition is more stable than that of the outdoor environment. Therefore, the appearance cue is more distinguishable than the spatio-temporal cue for target tracking.

The third environment is shown in Fig. 5(c). We installed eight cameras beside the roads of campus. We record a 7-h period in the daytime and set T_{max} and ΔT as 40 and 2 s, respectively. This environment is the most complex one, where the weak link problem is serious and difficult to be avoided simply by threshold selection.

Note that although first two environments have been also used in our early work [4], the true valid links were assumed to be given in [4]. In this study, the valid links are not given in advance and the weak links can be automatically discovered and removed.

B. Experiment on Learning Spatio-Temporal Relationship

In this section, we demonstrate the learning results and compare our method with that proposed by Makris *et al.* [23] for learning the spatio-temporal relationships. Notice that, although there are other methods [14], [32], [36] learning the spatio-temporal relationships automatically, the basic concept is based on Makris *et al.*'s method also.

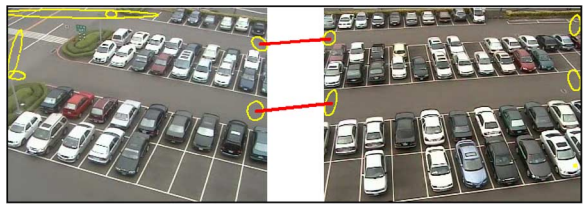

Fig. 6. Estimated entry/exit zones and valid links of the first environment.

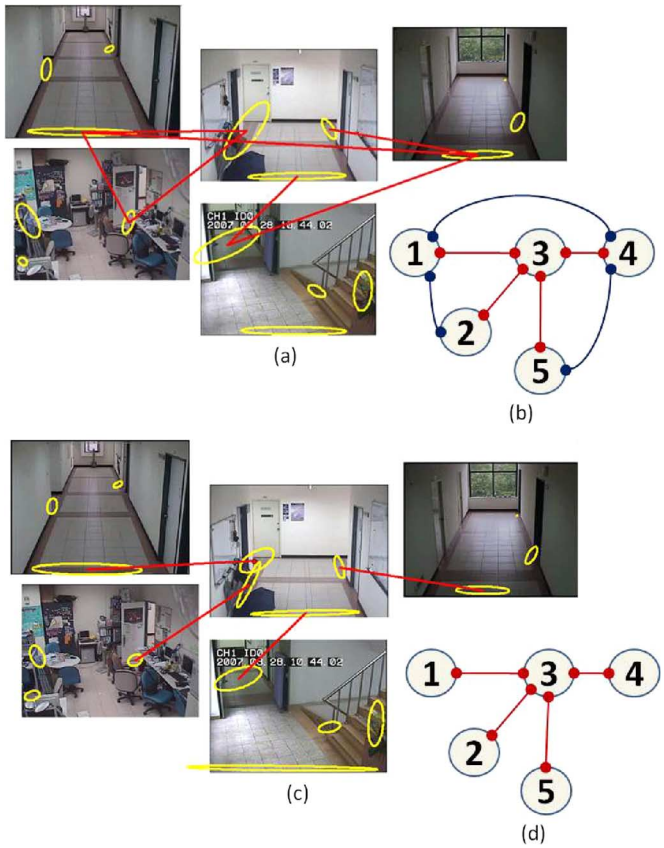

Fig. 7. Valid links and entry/exit zones of the second environment estimated by using (a) Makris *et al.* method and (c) our method. The corresponding camera topology estimated by using (b) Makris *et al.* method and (d) our method. The red lines represent the correct valid links, and the blue lines are weak links estimated.

In the first environment, both methods learn the correct valid links, and we show our result in Fig. 6 only. It demonstrates that both methods can work well in a simple environment. Fig. 7(a) and (b) shows the results of the second environment by using the method of [23] with 6-h batch training data. There are three weak links estimated. Fig. 7(c) and (d) shows the results estimated by our method, and we use the data during the first hour for batch learning and the other 5-h data for learning incrementally. The correct valid links and entry/exit zones without clustering fault are learnt. The incremental learning process is shown in the video sequence found at http://www.youtube.com/watch?v=VSIAQcMi3Nk. In this video, we use various lines to represent the strength of links, and the thick and red line means the valid links. It demonstrates that: 1) even lacking data when conducting batch learning, the truly valid links are estimated gradually by our incremental learning procedure, 2) it solves the clustering fault of entry/exit zones incrementally, and 3) it avoids the weak link problem. In

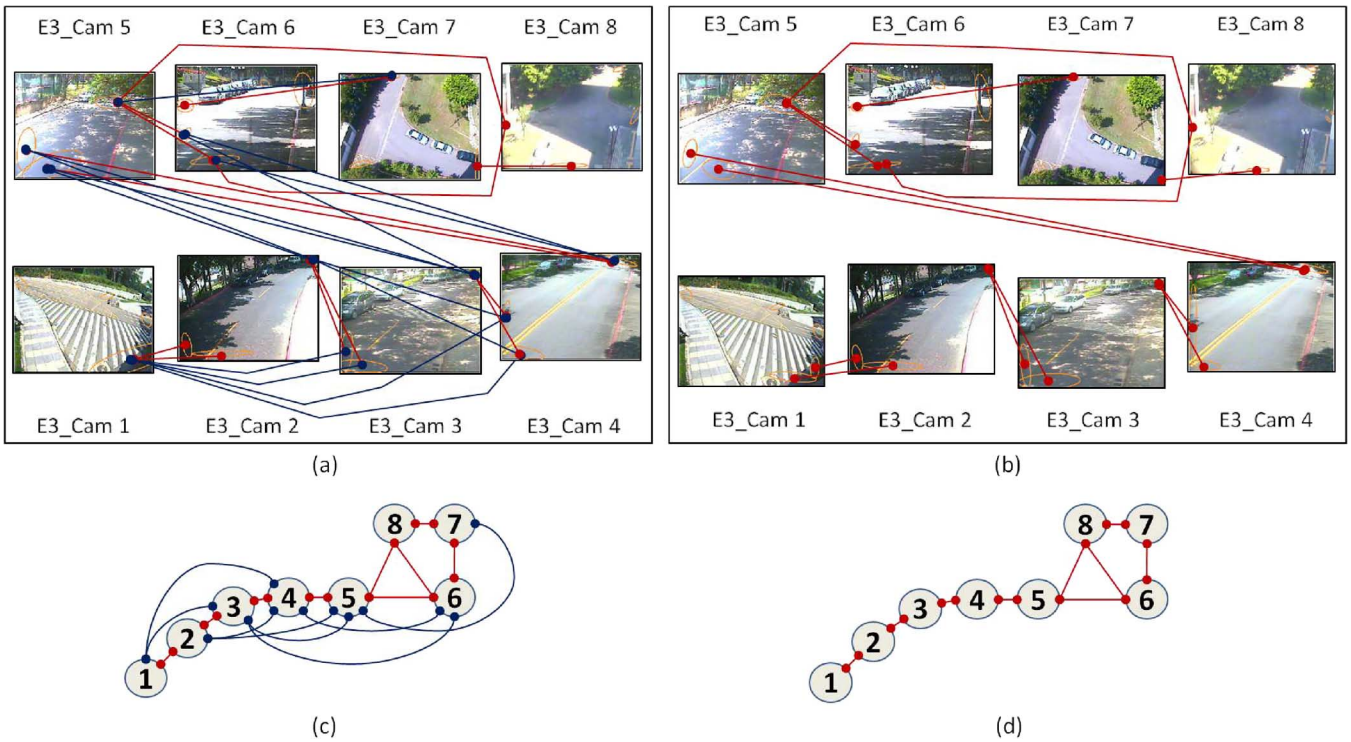

Fig. 8. Valid links and entry/exit zones of the thrid environment estimated by using (a) Makris *et al.* method and (b) our method. The corresponding camera topology estimated by using (c) Makris *et al.* method and (d) our method. The red lines represent the correct valid links, and the blue lines are weak links estimated.

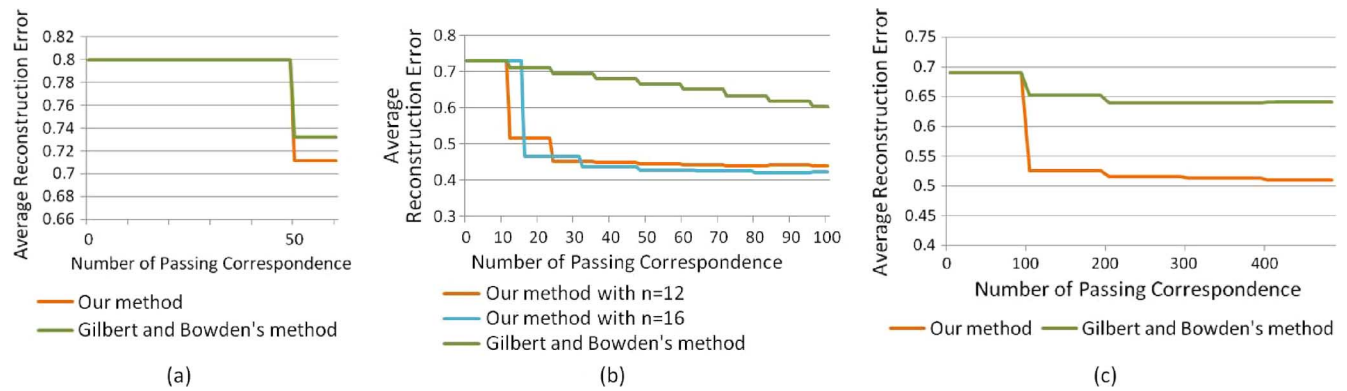

Fig. 9. Learning results of Gilbert and Bowden's method and our method. (a) Testing the camera pair of *E1_Cam 1* and *E1_Cam 2* in the first environment. (b) Testing the camera pair of *E2_Cam 3* and *E2_Cam 4* in the second environment. (c) Testing the camera pair of *E3_Cam 1* and *E3_Cam 2* in the third environment.

particular, in our early results [4] for the same environments, we avoid the weak link by assuming the true camera network topology has been provided manually in advance. Here, the weak link problem is overcome by the algorithm in Section V.

Then, both methods are applied to the video sequence of the third environment, as shown in Fig. 8. The method in [23] [Fig. 8(a) and (c)], with 7-h data for batch learning, suffers from the problem of weak links, where totally 14 weak links are mistaken for correct valid links. Fig. 8(b) and (d) shows the results estimated by our method with 1-h data for batch learning and 6-h data for increment learning. Our method learns all correct valid links and no weak links are misestimated.

C. Experiment on Learning Brightness Transfer Function

We compared our method with the method proposed by Gilbert and Bowden [14] for learning the appearance relationships. The appearance is modeled as a 256-bin histogram in this experiment. Their BTF is therefore a 256×256 matrix learnt by using the incremental learning procedure. Our BTF is learnt by the adaptive learning method described in Section VI with the dimension q equal to 15. We tested on each camera pair connected with a valid link, where the same correspondence determined by our spatio-temporal relationship is used for learning. We evaluated the learning results of BTF by calculating the average reconstruction error of hand-labeled ground truth data.

Fig. 9 shows some learning results, which belong to the camera pairs with the most different illumination in three environments. In Fig. 9(a), we show the result of the camera pair of *E1_Cam 1* and *E1_Cam 2* in the first environment. The results of both methods are similar, because the illuminations of both views of cameras are almost the same. In Fig. 9(b), we

show the result of the camera pair of *E2_Cam 3* and *E2_Cam 4*, and the period of passing 100 pairs of correspondence is about three hours. It shows that: 1) our method has a faster learning rate, and their method never learns a stable BTF in the testing period, 2) our method learns well by using few data and hence can re-build the appearance relationship models soon after sudden illumination changes, and 3) even with different number of collected matching data n (12 or 16), the results are converged after more data are collected. In Fig. 9(c), we show the result of the camera pair of *E3_Cam 1* and *E3_Cam 2*. Our method also learns well, and their method is difficult to converge to a lower value of the reconstruction error even when more than 500 targets have passed.

D. Experiment on Tracking Targets Across Multiple Cameras

We demonstrate the tracking results in three environments and compare four approaches: 1) baseline method, 2) only using spatio-temporal relationship, 3) baseline method combined with appearance relationship, and 4) using both relationships. The baseline method is that the entry/exit zones and the valid links, describing whether two zones are connected without knowing the transition time probability between zones, are determined manually. The appearance histograms are matched directly (i.e., without being transformed by BTF, and can be considered as the situation where the BTF is not learnt well) in the baseline method. The method using only spatio-temporal relationship, abbreviated as "*ST only*", is matching targets by using the transition time probabilities estimated by our method mentioned in Section IV. The baseline method refined by using appearance relationship, abbreviated as "*baseline + BTF*", is similar to the baseline method, but the appearance histograms are matched after transformation with the reconstructed BTF estimated by our method mentioned in Section VI. The last method, abbreviated as "*ST + BTF*", is learning both relationships automatically and combining them with a specified weighting factor, which is 0.6 and 0.4 in outdoor and indoor environments, respectively, for tracking. The tracking accuracy is defined as a ratio of the number of objects tracked correctly to the total number of objects passing through the scene.

In the first environment, the proposed system is trained for a 1-h period, and evaluated by using unseen ground-truth of half an hour. Totally 47 targets passed through the cameras during the testing data collection period of half an hour. In the second environment, the proposed system is trained for a 2-h period, and evaluated by using unseen ground-truth of half an hour. Totally 63 targets passed through the cameras during the testing data collection period of half an hour. In the third environment, the proposed system is trained for a 4-h period, and evaluated by using unseen ground-truth of an hour. Totally 798 targets passed through the cameras during the testing data collection period of an hour.

The overall tracking accuracy and the results of each pair of cameras are shown in Table I (the word in red color representing the best result). It is obvious that using either spatio-temporal relationship or appearance relationship improves the tracking accuracy compared with the baseline method, and combining both cues, i.e., *ST + BTF*, will lead to the best results. It concludes that our method performs well and achieves high tracking accuracy in both indoor and outdoor environments. Some tracking results can be seen in the video sequence found at http://www.youtube.com/watch?v=XZDf6WU1pmI.

TABLE I
TRACKING ACCURACY

Environment #1	Tracking Accuracy			
Camera Pair Number	baseline method	ST only	baseline + BTF	ST + BTF
1 - 2 (Overall)	59.60%	89.40%	63.80%	89.40%
Environment #2	**Tracking Accuracy**			
1 - 3	65.20%	86.90%	82.60%	86.90%
2 - 3	78.90%	63.20%	94.70%	94.70%
3 - 4	54.50%	90.90%	100%	100%
3 - 5	80%	60%	90%	90%
Overall	69.80%	76.20%	90.40%	92.10%
Environment #3	**Tracking Accuracy**			
1 - 2	74.51%	81.70%	77.78%	81.70%
2 - 3	77.78%	91.50%	87.58%	91.50%
3 - 4	73.65%	81.08%	77.03%	81.76%
4 - 5	80.65%	83.23%	82.58%	85.81%
5 - 6	65.71%	72.86%	62.86%	74.29%
5 - 8	42.86%	57.14%	42.86%	71.43%
6 - 7	89.47%	89.47%	84.21%	89.47%
6 - 8	22.22%	33.33%	22.22%	33.33%
7 - 8	42.86%	92.86%	64.29%	92.86%
Overall	73.56%	81.83%	76.82%	82.83%

E. Discussion

From Table I, we observe that the appearance relationship improved tracking more than spatio-temporal relationship in an indoor environment. On the contrary, the spatio-temporal relationship got better results in outdoor environments. The reasons are as follows. First, the persons sometimes walk together, but the vehicles usually keep the distance, so that the spatio-temporal relationship can get a better result when tracking vehicles. Second, the appearance of different persons is usually more distinct, but the appearances of vehicles are sometimes very similar. In our experiments, the difference of average appearance match errors for indoor pedestrians between correct match and false match is about 0.12, but it is only about 0.02 for outdoor vehicles. The reason is that we use the color feature only and the vehicle colors are limited. According to the website [2], 90% of vehicle colors in North America in 2008 are listed below: white, black, silver, blue, gray, and red.

In Section VII-A, we have introduced the special property of the first environment, a parking lot, where the transition time is varied in different situations. We compare our incremental learning method with the baseline method and the batch learning methods that use two different learning periods, which are from 0 to 30 min and from 30 to 60 min, respectively. Then we evaluate them by using unseen ground-truth of half an hour. Notice that the period for evaluation is the video sequence from 60 to 90 min, where the parking lot is almost full. The tracking accuracy of four approaches (baseline method, 30 min batch learning by the video sequence from 0 to 30 min, 30 min batch learning

by the video sequence from 30 to 60 min, and our incremental learning method) are 59.6%, 68.09%, 89.40%, and 89.4%, respectively. It shows that the results of batch learning are much distinct by using different learning periods. This is also the main problem of batch learning methods for which we always do not know when to learn, and how long the learning period is necessary. On the contrary, the proposed method is better for different situations and long-term monitoring.

VIII. Conclusion

Unlike other approaches assuming that the monitored environments remain unchanged, we have presented an adaptive and unsupervised method for learning both spatio-temporal and appearance relationships for a camera network. It can incrementally refine the clustering results of the entry/exit zones and the transition time probability distributions, and learns the subspace of BTFs in a short period of time by combing the spatio-temporal information and efficient MCMC sampling. Two common problems, the weak link problem and illumination-change problem, are solved, which have never been studied by previous works.

References

[1] Q. Cai and J. K. Aggarwal, "Tracking human motion in structured environments using a distributed-camera system," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1241–1247, Nov. 1999.

[2] cars.com. [Online]. Available: http://www.cars.com., 2008

[3] K. W. Chen and Y. P. Hung, "Multi-cue integration for multi-camera tracking," in *Proc. Int. Conf. Pattern Recognition*, 2010, pp. 145–148.

[4] K. W. Chen, C. C. Lai, Y. P. Hung, and C. S. Chen, "An adaptive learning method for target tracking across multiple cameras," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[5] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proc. IEEE*, vol. 89, no. 10, pp. 1456–1477, Oct. 2001.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, pp. 142–149.

[7] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1999.

[8] F. Dellaert, Addressing the Correspondence Problem: A Markov Chain Monte Carlo Approach, Carnegie Mellon Univ., Sch. Comput. Sci., Pittsburgh, PA, 2000, Tech. Rep.

[9] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. B-39, no. 1, pp. 1–38, 1977.

[10] H. Detmold, A. Hengel, A. Dick, A. Cichowski, and R. Hill, "Topology estimation for thousand-camera surveillance networks," in *Proc. ACM/IEEE Int. Conf. Distributed Smart Cameras*, 2007, pp. 195–202.

[11] A. Dick and M. Brooks, "A stochastic approach to tracking objects across multiple cameras," in *Proc. Australian Conf. Artificial Intelligence*, 2004, pp. 160–170.

[12] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 2360–2367.

[13] N. Gheissari, T. B. Sebastian, P. H. Tu, and J. Rittscher, "Person reidentification using spatiotemporal appearance," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 1528–1535.

[14] A. Gilbert and R. Bowden, "Incremental, scalable tracking of objects inter camera," *Comput. Vis. Image Understand.*, vol. 111, no. 1, pp. 43–58, 2008.

[15] A. Girgensohn, F. Shipman, T. Turner, and L. Wilcox, "Effects of presenting geographic context on tracking activity between cameras," in *Proc. ACM Conf. Computer-Human Interaction*, 2007, pp. 1167–1176.

[16] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[17] T. Huang and S. Russell, "Object identification in a Bayesian context," in *Proc. Int. Joint Conf. Artificial Intelligence*, 1997, pp. 1276–1282.

[18] O. Javed, K. Shafique, Z. Rasheed, and M. Shah, "Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views," *Comput. Vis. Image Understand.*, vol. 109, no. 2, pp. 146–162, 2008.

[19] C. H. Kuo, C. Huang, and R. Nevatia, "Inter-camera association of multi-target tracks by on-line learned appearance affinity models," in *Proc. Eur. Conf. Computer Vision*, 2010, pp. 383–396.

[20] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1999, pp. 252–259.

[21] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1355–1360, Oct. 2003.

[22] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *IEEE Trans. Syst., Man., Cybern. B*, vol. 35, no. 3, pp. 387–408, Jun. 2005.

[23] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004, pp. 205–210.

[24] A. Mittal and D. Huttenlocher, "Scene modeling for wide area surveillance and image synthesis," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, pp. 160–167.

[25] H. T. Nguyen and A. W. M. Smeulders, "Multiple target tracking by incremental probabilistic PCA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 52–64, Jan. 2007.

[26] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man., Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[27] H. Pasula, S. Rusell, M. Ostland, and Y. Ritov, "Tracking many objects with many sensors," in *Proc. Int. Joint Conf. Artificial Intelligence*, 1999, pp. 1160–1171.

[28] F. Porikli and A. Divakaran, "Multi-camera calibration, object tracking and query generation," in *Proc. Int. Conf. Multimedia and Expo*, 2003, pp. 653–656.

[29] B. Prosser, S. Gong, and T. Xiang, "Multi-camera matching using bi-directional cumulative brightness transfer functions," in *Proc. British Machine Vision Conf.*, 2008.

[30] K. Shafique, A. Hakeem, O. Javed, and N. Haering, "Self calibrating visual sensor networks," in *Proc. IEEE Workshop Applications of Computer Vision*, 2008, pp. 1–6.

[31] M. Song and H. Wang, "Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering," in *Proc. SPIE Conf. Intelligent Computing: Theory and Applications III*, 2005, pp. 174–183.

[32] C. Stauffer, "Learning to track objects through unobserved regions," in *Proc. IEEE Workshop Motion and Video Computing*, 2005, pp. 96–102.

[33] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[34] C. Stauffer and K. Tieu, "Automated multi-camera planar tracking correspondence modeling," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003, pp. 259–266.

[35] M. J. Swain and D. H. Ballard, "Indexing via color histograms," in *Proc. IEEE Int. Conf. Computer Vision*, 1990, pp. 390–393.

[36] K. Tieu, G. Dalley, and W. Grimson, "Inference of non-overlapping camera network topology by measuring statistical dependence," in *Proc. IEEE Int. Conf. Computer Vision*, 2005, pp. 1842–1849.

[37] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Roy. Statist. Soc., Series B*, vol. 61, no. 3, pp. 611–622, 1999.

[38] X. Wang, K. Tieu, and W. E. L. Grimson, "Correspondence-free multi-camera activity analysis and scene modeling," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[39] B. Xie, V. Ramesh, and T. E. Boult, "Sudden illumination change detection using order consistency," *Image Vis. Comput.*, vol. 22, no. 2, pp. 117–125, 2004.

Kuan-Wen Chen received the B.S. degree in computer and information science from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and the Ph.D. degree in computer science and information engineering at National Taiwan University, Taipei, Taiwan, in 2011.

His current research interests include computer vision, pattern recognition, visual surveillance, and human–computer interaction.

Chih-Chuan Lai received the B.S. degree in electronic engineering and the M.S. degrees in computer science and information engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 1999 and 2001, respectively. He is currently pursuing the Ph.D. degree in the Institute of Networking and Multimedia at National Taiwan University.

His current research interests include pattern recognition, image processing, and computer vision.

Pei-Jyun Lee received the B.S. degree in computer science and information engineering from National Dong Hwa University, Taipei, Taiwan, in 2007 and the M.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2010.

She is currently a senior engineer at Mstar Semiconductor, Inc., Taiwan. Her research interests include computer vision, visual surveillance, and human–computer interaction.

Chu-Song Chen (S'94–M'96) received the B.S. degree in control engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1989 and the M.S. and Ph.D. degrees from the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, in 1991 and 1996, respectively.

He is now a deputy director of the Research Center for Information Technology Innovation (CITI), and a research fellow of the Institute of Information Science (IIS), Academia Sinica, Taipei. He is also an adjunct professor of the Graduate Institute of Networking and Multimedia, National Taiwan University. His research interests include computer vision, pattern recognition, signal/image processing, and multimedia analysis. In 2007-2008, he served as the Secretary-General of the IPPR Society, Taiwan, which is one of the regional societies of the International Association of Pattern Recognition (IAPR).

Dr. Chen has served as the program co-chairs of the conferences ICDAT 2005 and ICDAT 2006, theme chair of PSIVT 2009, and area chairs of ACCV 2009, ACCV 2010, and NBiS 2010. He is on the editorial board of *Journal of Multimedia* (Academy Publisher), *Machine Vision and Applications* (Springer), and *IPSJ Transactions on Computer Vision and Applications*.

Yi-Ping Hung (S'84-M'89) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1982, the M.S. degree from the Division of Engineering, Brown University, Providence, RI, the M.S. degree from the Division of Applied Mathematics, Brown University, and the Ph.D. degree from the Division of Engineering, Brown University, in 1987, 1988, and 1990, respectively.

He is currently a Professor with the Graduate Institute of Networking and Multimedia and with the Department of Computer Science and Information Engineering, National Taiwan University. From 1990 to 2002, he was with the Institute of Information Science, Academia Sinica, Taipei, where he became a tenured Research Fellow in 1997 and is currently a Joint Research Fellow. He has served as the Deputy Director of the Institute of Information Science from 1996 to 1997, and the Director of the Graduate Institute of Networking and Multimedia, National Taiwan University, since 2007. His current research interests include computer vision, pattern recognition, image processing, virtual reality, multimedia, and human–computer interaction.

Dr. Hung was the Program Cochair of ACCV'00 and ICAT'00 and the Workshop Cochair of ICCV'03. He has been an Editorial Board Member of the *International Journal of Computer Vision* since 2004.