Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2009, Article ID 460689, 19 pages
doi:10.1155/2009/460689

Research Article

Continuous Learning of a Multilayered Network Topology in a Video Camera Network

Xiaotao Zou, Bir Bhanu, and Amit Roy-Chowdhury

Center for Research in Intelligent Systems, University of California, Riverside, CA 92521, USA

Correspondence should be addressed to Xiaotao Zou, xzou@ee.ucr.edu

Received 20 February 2009; Revised 18 June 2009; Accepted 23 September 2009

Recommended by Nikolaos V. Boulgouris

A multilayered camera network architecture with nodes as entry/exit points, cameras, and clusters of cameras at different layers is proposed. Unlike existing methods that used discrete events or appearance information to infer the network topology at a single level, this paper integrates face recognition that provides robustness to appearance changes and better models the time-varying traffic patterns in the network. The statistical dependence between the nodes, indicating the connectivity and traffic patterns of the camera network, is represented by a weighted directed graph and transition times that may have multimodal distributions. The traffic patterns and the network topology may be changing in the dynamic environment. We propose a Monte Carlo Expectation-Maximization algorithm-based continuous learning mechanism to capture the latent dynamically changing characteristics of the network topology. In the experiments, a nine-camera network with twenty-five nodes (at the lowest level) is analyzed both in simulation and in real-life experiments and compared with previous approaches.

1. Introduction

Networks of video cameras are being envisioned for a variety of applications and many such systems are being installed. However, most existing systems do little more than transmit the data to a central station where it is analyzed, usually with significant human intervention. As the number of cameras grows, it is becoming humanly impossible to analyze dozens of video feeds effectively. Therefore, we need methods that can automatically analyze the video sequences collected by a network of cameras.

Most work in computer vision has concentrated on a single or a few cameras. While these techniques may be useful in a networked environment, more is needed to analyze the activity patterns that evolve over long periods of time and large swaths of space. To understand the activities observed by a multicamera network, the first step is to infer the spatial organization of the environment under surveillance, which can be achieved by camera node localization [1], camera calibration [2, 3], or camera network topology inference [4–7] for different purposes. In this paper, we focus on the topology inference of the camera network consisting of cameras with mostly nonoverlapping field-of-views (FOVs).

Similar to the notion used in computer networking community, the camera network topology is the study of the arrangement or mapping of the nodes in a camera network [8]. There are two main characteristics of network topology: firstly, the existence of possible links between nodes (i.e., the connectivity), which correspond to the paths that can be followed by objects in the environment; secondly, the transition time distribution of pedestrians observed over time for each valid link ("path"), which is analogous to the latency studied in the communication networks. Rather than learning the geometrically accurate maps by networked camera localization [1], the objective of topology inference is to determine the topological map of the nodes in the environment. The applications of the inferred camera network topology may include coarse localization of the networked cameras, anomalous activity detection in a multi-camera network, and multiple object tracking in a network of distributed cameras with non-overlapping FOVs.

In this paper we develop (i) a multi-layered network architecture that allows analysis of activities at various resolutions, (ii) a method for learning the network topology in an unsupervised manner by integrating visual appearance

and identity information, and (iii) a Markov Chain Monte Carlo (MCMC) learning mechanism to update the network topology framework continuously in a dynamically changing environment. The paper does not deal with how to optimally place these cameras; it focuses on how to infer the connectivity and further analyze activities given fixed locations of the cameras. We now highlight the relation with the existing work and the main contributions of this paper along these lines.

Section 2 describes the related work and contributions of this paper. The multi-layered network architecture is described in Section 3.1. In Section 3.2, we present our theory for learning the network topology by integrating identity and appearance information, followed by the approach for identifying network traffic patterns. In Section 4, we first show extensive simulation results for learning a multi-layered network topology and for activity analysis; then, experimental results in a real-life environment are presented. Finally, we conclude the paper in Section 5.

2. Related Work and Contributions

Camera network is an interdisciplinary area encompassing computer vision, sensor networks, image and signal processing, and so forth. Thanks to the mass production of CCD or CMOS cameras and the increasing requirement in elderly assistance, security surveillance and traffic monitoring, a large number of video camera networks have been deployed or are being constructed in our every-day life. In 2004, it was estimated [9] that the United Kingdom was monitored by over four million cameras, with practically all town centers under surveillance. One of the prerequisites for processing and analyzing the visual information provided that randomly placed sensors is to generate the spatial map of the environment. In the sensor networks and computer vision communities, there has been a large body of work on network node localization or multi-camera self-calibration. In most cases, the node localization/calibration involves the discovery of location information and/or the orientation information (in the case of cameras) of the sensor nodes.

In the research by Fisher [3], it was shown that it is possible to solve the calibration problem for the randomly placed visual sensors with non-overlapping field-of-views. It presented a possible solution by using distant objects to recover orientation and nearby objects to recover relative locations. However, it employed a strict assumption on the motion of the observed objects. Ihler et al. [10] presented nonparametric belief propagation-based self-calibration method from pairwise distance estimates of sensor nodes. Inspired by the success of Simultaneous Localization and Mapping (SLAM) [11] in robot navigation, Simultaneous Localization And Tracking (SLAT) [1, 2] was proposed and widely used in sensor network. SLAT is to calibrate and localize the nodes of a sensor network while simultaneously tracking a moving target observed by the sensor network. Rahimi et al. [2] proposed a probabilistic model-based optimization algorithm to address the SLAT problem, which computed the most likely trajectory and the most likely calibration parameters with the Newton-Raphson method. Rather than the offline and centralized algorithm in [2], Funiak et al. [1] used the Boyen and Koller algorithm which is an approximation to the Kalman filtering as the basis and built a scalable distributed filtering algorithm to solve the SLAT problem.

The geometric maps, generated by SLAT, can be used for reliably mapping the observations from sensor nodes to the global 2D ground-plane or 3D space coordinate system of the environment. For a large number of applications, however, the topological map is more suitable and more efficient than the geometric map. For example, the human activity analysis presented by Makris and Ellis in [12] was based on trajectory observations and *a priori* knowledge of the network topology. This provided an understanding of the paths that can be followed by objects within the field of view of the network of cameras.

Javed et al. [13] presented a supervised learning algorithm to simultaneously infer the network topology and track objects across non-overlapping field-of-views. They employed a Parzen window technique that looks for correspondences in object velocity, intercamera transition time, and the entry/exit points of objects in the FOV of a camera. However, the work in [13] relies on the strict constraint of manually labeled trajectories, which is costly and not always available in the real environment. With respect to the wide use of non-overlapping cameras in camera networks, there is the need for new methods to relax the assumption of known data correspondence.

Recently, there has been some work on understanding the topology of a network of non-overlapping cameras [5, 6, 14] and using this to make inferences about activities viewed by the network [12]. The authors in these papers proposed an interesting approach for modeling activities in a camera network. They defined the entry/exit points in each camera as nodes and learned the connectivity between these nodes. Makris et al. [4] proposed a cross correlation-based statistical method to capture the temporal correlation of departures and arrivals of objects in the field-of-views, which in turn is used to infer the network topology with unknown correspondence. Tieu et al. [14] used the information theoretic-based statistical dependence to infer the camera network topology, which integrated out the uncertain correspondence using Markov Chain Monte Carlo (MCMC) method [15].

Marinakis et al. [6] used the Monte Carlo Expectation-Maximization (MC-EM) algorithm to simultaneously solve the data correspondence and network topology inference problems. The MC-EM algorithm [16, 17] expands the scope of the EM by executing the Expectation step, which is intractable to sum over the huge volume of unknown data correspondence, through MCMC sampling. This approach works well for a limited number of moving objects (e.g., mobile robots) observed by the sensor network. When data correspondence for a large number of objects is encountered, the number of samples in MC-EM algorithm will increase accordingly, which makes the convergence of MCMC sampling to the correct correspondence really slow.

(a) *A* in camera 1 (b) *A* in camera 2 (c) *B* in camera 2

Figure 1: An example of false appearance similarity information. Two subjects ("*A*" and "*B*") are monitored by two cameras ("1" and "2"). Their clothing is similar, and the illumination of these two cameras is different. The Bhattacharyya distances between the RGB color histograms of the extracted objects in the above three frames ("a," "b," and "c") are calculated to identify the objects: d(a,b) = 0.9097, and d(a, c) = 0.6828, which will establish a false correspondence between "a" and "c. "

All these approaches take only the discrete "departure/arrival" time sequences as input. To employ the abundant visual information provided by the imaging sensors, Niu and Grimson [5] proposed an appearance-integrated cross-correlation model for topology inference on the vehicle tracking data. It computed the appearance similarity of objects at departure and arrivals as the product of the normalized color similarity and size similarity. However, appearances (e.g., color) may be deceiving in real-life applications. For example, clothing color of different human subjects is similar ("false match") as shown in Figures 1(a) and 1(c), or cloth color of the same object changes significantly under different illuminations ("false nonmatch") in Figures 1(a) and 1(b). Besides, it is hard to differentiate human subjects based on the observed size observed in the overhead cameras.

Furthermore, these approaches work in a "one-shot" manner; that is, once the topology is inferred, it is assumed not to change. However, the assumption cannot be guaranteed in the dynamic changing environment. The traffic behaviors in such environment vary much depending on the age, health status, and so forth of the pedestrians. Besides, the nature of the pan-tilt-zoom cameras widely used in the sensor networks renders the "static environment" assumption invalid. These issues prompt a continuous learning framework for camera network topology inference as presented in our paper.

We compare our approach and the existing work in network topology inference in Table 1. Both transition times and face recognition are helpful and used in our work. We are not aware of any other published approach that has used both transition times and face recognition. This information can also be useful for anomaly detection in a video network. The author in [18] explores the joint space of time delay and face identification results for the detection of anomalous behavior.

We propose a principled approach to integrate the appearance and identity (e.g., face) to enhance the statistics-based network topology inference. The main contributions of the paper are summarized in the following.

(A) Multilayered Network Architecture. The work in [5, 14] defines the network as a weighted graph linking different nodes defined by the entry/exit points in the cameras. The links in the graph define the permissible paths. If a user were presented with just this model, he/she would have to do a significant amount of work to understand the connectivity between all the cameras. However, applications may demand that we model only the paths between the cameras without regard to what is happening within the field-of-views (FOV) of individual cameras. This means that we need to cluster the nodes into groups based on their location in each camera. Taking this further, we can cluster the cameras into groups. For example, if there are hundred cameras in the whole campus, we may want to group them depending upon their geographical location. This is the motivation for our multilayered network architecture.

At the lowest level the connectivity is between the nodes defined by entry/exit points. At the higher level, we cluster these nodes based on their location within the FOV of each camera. At the third level, the cameras are grouped together. This can continue depending upon the number of cameras, their relative positions, and the application. (An example of a multilevel architecture is given in Figure 3.) At each level, we learn the network topology in an unsupervised manner by observing the patterns of activities in the entire network. Note that given the information at the highest resolution (i.e., at the lowest level), we can get the network graphs at the upper levels, but not vice versa.

Departure and arrival locations in each camera view are nodes in the network at the lowest level of the architecture (see Figure 3). A link between a departure node and an arrival node denotes connectivity. By topology we mean to determine which links exist. The links are directional and they can be bidirectional. The information about the identities is stored at the nodes corresponding to entry/exit points at the bottom level of the network architecture.

(B) Integrating Appearance and Identity for Learning Network Topology. The work in [5] uses the similarity in appearance

Table 1: A comparison of our approach with the state-of-the-art topology inference approaches suited for non-overlapping camera networks.

Approaches	Makris et al. [4]	Tieu et al. [14]	Marinakis et al. [6]	Niu and Grimson [5]	Our approach
Method	Cross correlation	MCMC and Mutual information	Monte Carlo Expectation-Maximization	Appearance-weighted cross correlation	Weighted cross correlation and MC-EM
Continuous learning?	NO	NO	NO	NO	YES
Input	Discrete departure/arrival sequence (D/A)	Discrete D/A	Discrete D/A	Discrete D/A and appearance	Discrete D/A, appearance and identity
Visual cues	N/A	N/A	N/A	Appearance	Appearance and identity
Node level	Single (entry/exit points)	Single (entry/exit points)	Single (entry/exit points)	Single (entry/exit points)	3-level (entry/exit points, cameras and camera clusters)
Link validation	Threshold-ing	Mutual information	Posterior probability	Mutual information	Mutual information
Camera orientation	N/A	Overhead and side-facing	N/A	Side-facing	Overhead and side-facing
Complexity of simulation	N/A	22 nodes	80 directed links in 20 nodes	26 nodes	25 nodes in 9 cameras
Complexity of real experiments	26 nodes in 6 cameras	15 nodes in 5 cameras	7 nodes in 6 cameras	10 nodes in 2 cameras	25 nodes in 9 cameras and 13 links
Performance evaluation	NO	YES	YES	YES	YES

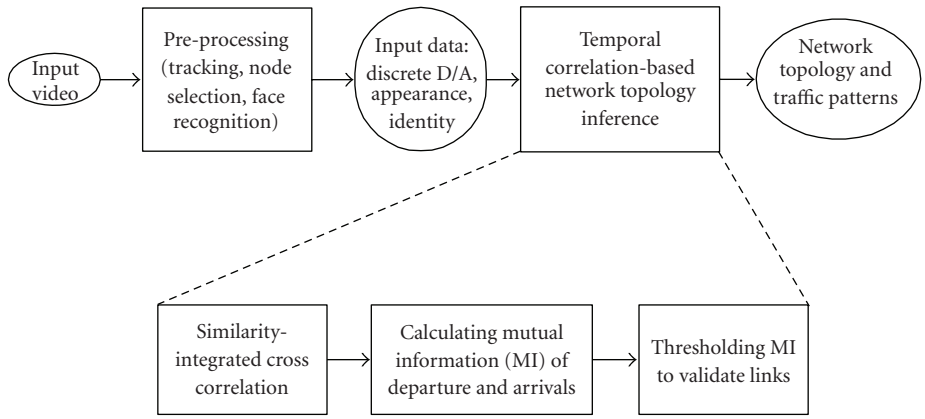

Figure 2: The block diagram of the proposed method.

to find correlations between the observed sequences at different nodes. However, appearances may be deceiving in many applications as in Figure 1. For this purpose, we integrate human identity (e.g., face recognition in our experiments) whenever possible in order to learn the connectivity between the nodes. We provide a principled approach for doing this by using the joint distribution of appearance similarity and identity similarity to weight the cross-correlation. We show through simulations and real-life examples how adding identity can improve the performance significantly over existing methods.

Note that the identity information can be very useful for learning network topology since the color information alone is not reliable. However, face recognition is not the focus of this paper. Existing techniques for frontal face recognition [19–21] or side face recognition [22] in video can provide improved performance. For a network of video cameras, see [23, 24] and for intercamera tracking, see [25].

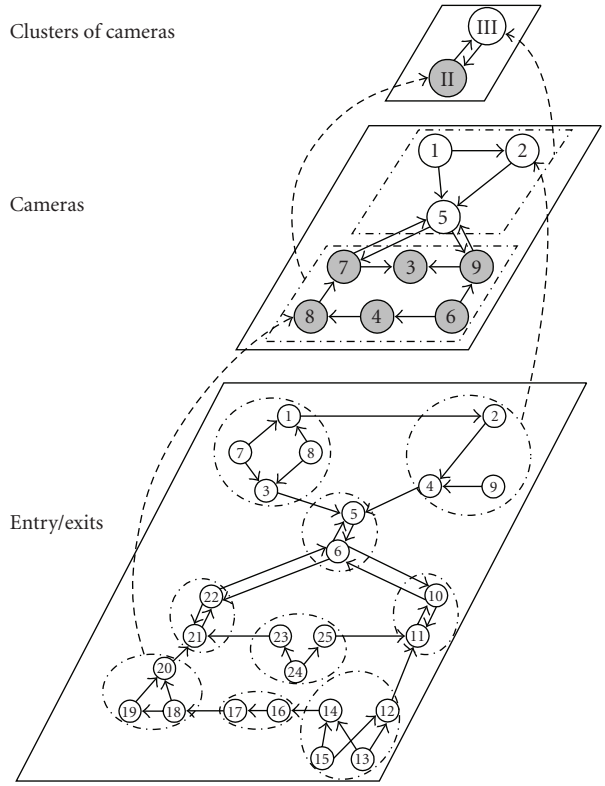

Figure 3: The three-layered architecture of the camera network.

(C) Continuous Learning of Traffic Patterns and Network Topology in the Dynamically Changing Environment. As shown in Table 1 the previous work only focuses on the batch-mode learning of traffic patterns and network topology in the static environment. However, the traffic patterns and the network topology keep changing in the dynamic environment. The continuous learning mechanism proposed in the paper is necessary for the topology inference to reflect the latent dynamically changing characteristics.

3. Technical Approach

The technical approach proposed in this paper consists of a multi-layered network architecture, the inference of network topology and traffic patterns, and the continuous learning of the network topology and traffic patterns in the dynamically changing environment. The block diagram of the system is shown in Figure 2.

3.1. Multilayered Network Architecture. The network topology is defined as the connectivity of nodes in the network. For instance, given the node as a single camera in a distributed camera network as in [6], the network topology is the connectivity of all the cameras in the network. In [5, 14], the entry/exit points are defined as the nodes in the network and a weighted directed graph is employed to represent the network topology. The advantage of "entry/exit" nodes is the detailed description of the network topology. The disadvantage of such representation is the cumbersome volume of the network to analyze. For instance, a network with 9 cameras will give rise to at least 18 entry/exit points as nodes, which may have up to 306 directed links.

To deal with the increasing number of cameras installed for surveillance nowadays, we propose a multi-layered architecture of weighted, directed graphs as the camera network topology (as shown in Figure 3), which can maintain scalability and granularity for analysis purposes. Figure 3 is actually the network architecture for our experimental setup and the simulation, which will be described in Section 4 in detail.

In the hierarchical architecture in Figure 3, the nodes at the lowest level are the entry/exit points in the FOVs of cameras; the middle level is composed of the nodes as single cameras; the top level has the fewest nodes that correspond to the clusters of cameras, for example, all the cameras on the second (II) and third (III) floors of a building, respectively. All the entry/exit points in the same FOV can be grouped and associated with the corresponding camera node at the middle level. Similarly, the camera nodes in the middle level can be grouped according to their geographic locations and associated to the appropriate node at the highest "*cluster*" level. For example, in Figure 3, the entry/exit nodes "18," "19," and "20" are in the FOV of the camera "8," which is associated with the cluster "II" along with other cameras on the same floor.

The topology is inferred in a bottom-up fashion: first at the lowest "*entry/exit*" level, then at the middle "*camera*" level, and finally at the highest "*cluster*" level. In subsequent network traffic pattern analysis, the traffic can be analyzed at the "*entry/exit*" level, at the "*camera*" level, or even at the "*cluster*" level, if applicable, which provides a flexible scheme for traffic pattern analysis at various resolutions. Note that since the single layer network deals only with the entry/exit patterns, the computational burden will be the same in a single-layer network and the bottom layer of the multi-layer network. Multi-layer network architecture processes data at a lower level and the information is passed to a higher level. It requires more computational resources since higher-level associations need to be formed. However, the hierarchical architecture allows, if desired, the passing of control signals in a top down manner for active control of network cameras.

3.2. Inferring Network Topology and Identifying Traffic Patterns. In this section, we will show how to determine the camera network topology by measuring the statistical dependence of the nodes with the appearance and identity (when available); then the topology inference for the multi-layered architecture and the network traffic pattern identification are presented. Finally, continuous learning of traffic patterns and network topology is described.

3.2.1. Inference of Network Topology. The network topology is inferred in a bottom-up fashion. We first show how to infer the topology at the "*entry/exit*" level by integrating appearance and identity. At the lowest level of our multi-layered network architecture, the nodes denote the entry/exit points in the FOVs of all cameras in the network. They can be manually chosen or automatically set by clustering the ends of object trajectories. If they are in the same FOV or in the overlapping FOVs, it is easy to infer the connectivity between them by checking object trajectories through the views. In this paper, we focus on the inference of connectivity between nodes in non-overlapping FOVs, which are blind to the cameras. The network topology at the lowest level is represented by a weighted, directed graph with nodes as entry/exit points and the links indicating the connectivity between nodes.

Suppose that we are checking the link from node i to node j. We observe objects departing at node i and arriving at node j. The departure and arrival events are represented as temporal sequences $X_i(t)$ and $Y_j(t)$, respectively. We define $A_{X,i}(t)$ and $A_{Y,j}(t)$ as the observed appearances in the departure and arrival sequences, respectively. The identities of the objects observed at the departure node i and at the arrival node j are $I_{X,i}(t)$ and $I_{Y,j}(t)$, respectively.

Niu and Grimson [5] present an appearance similarity-weighted cross correlation method to infer the connectivity of nodes. To alleviate the sole dependence on appearance, which is deceiving when the objects are humans, we propose to use the appearance and identity information to weigh the statistical dependence between different nodes, that is, the cross-correlation function of departure and arrival $X_i(t)$ and $Y_j(t)$:

$$R_{i,j}(\tau) = E\Big[X_i(t)\cdot Y_j(t+\tau)\Big] = \sum_{t=-\infty}^{\infty} X_i(t)\cdot Y_j(t+\tau) = E\Big[f\Big(A_{X,i}(t), A_{Y,j}(t+\tau), I_{X,i}, I_{Y,j}(t+\tau)\Big)\Big], \tag{1}$$

where f is the statistical similarity model of appearances and identity, which implicitly indicates the correspondence between subjects observed in different views. The joint model of f and its components are presented in the following subsections. An example is given in Figure 4. From now on, we assume that departure and arrival nodes are always i and j, respectively, so that the subscripts i and j can be omitted.

3.2.2. Statistical Model of Identity. The working principles of the human identification are as follows: (1) detect the departure/arrival objects and employ image enhancement techniques if needed (e.g., the superresolution method for face recognition); (2) the objects departing from node i are represented by unique identities $I_X(t)$, which are used as the gallery; (3) the identities $\widetilde{I}_Y$ of the objects arriving at the node j are identified by comparing it with all objects in the gallery, that is,

$$S_{\mathrm{ID}}\left(\widetilde{I}_Y\right) = \arg\max_{I_X}(\mathrm{sim}(I_Y, I_X)), \tag{2}$$

where $\mathrm{sim}(I_Y, I_X)$ is the similarity score between I_Y and I_X, and $S_{\mathrm{ID}}(\cdot)$ is the similarity score of the identified identity.

We use the mixture of Gaussian distributions (e.g., as shown in Figure 5) to model the similarity scores of identities:

$$P_{\mathrm{ID}} = P\left(S_{\mathrm{ID}}\left(\widetilde{I}_Y\right) \mid X = Y\right) = \sum_{m=1}^{k} a_m \cdot N(\mu_m, \sigma_m^2), \tag{3}$$

where k is the number of components, α_m is the weights, μ_m and σ_m^2 are the mean and variance of the mth Gaussian component, and $X = Y$ means that they correspond to the same object.

The unknown parameters $\{k, \alpha_m, \mu_m, \text{and } \sigma_m^2\}$ can be estimated by using the Expectation-Maximization (EM) algorithm [26] in face recognition experiments on large datasets. The mixture of Gaussians in Figure 5, which has four components, is obtained by using EM algorithm in the identification experiments [27].

3.2.3. Statistical Model of Appearance Similarity. We employ the comprehensive color normalization (as in [5]) to alleviate the dependence of appearances on the illumination condition. Then, the color histograms in the hue and saturation space, that is, h and s, respectively, are calculated on the normalized appearance. Note that we do not incorporate the size information in the appearance metrics because the observed objects are humans. We first normalize the sizes

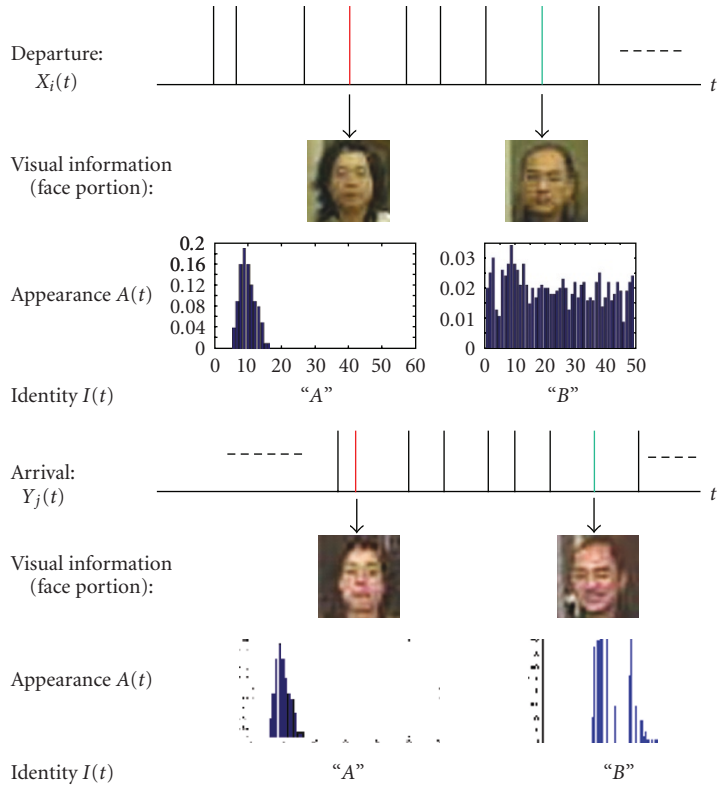

Figure 4: An example of observed "departure/arrival" sequences and corresponding appearance (as the normalized color histogram)and identities for two distinct subjects.

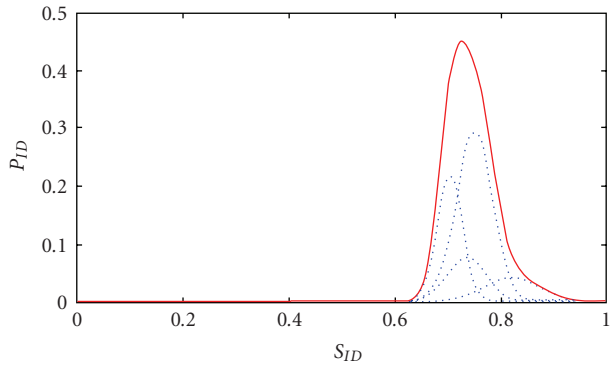

Figure 5: The Gaussian mixture model of the identity similarity.

(i.e., heights and widths) of objects before calculating color metrics.

Next, a multivariate Gaussian distribution $(N(\mu_{h,s}, \Sigma_{h,s}))$ is fitted to the color histogram similarity between the two appearances:

$$P_{\text{app}} = P\left(h_X - h_y, s_X - s_Y \mid X = Y\right) \sim N(\mu_{h,s}, \Sigma_{h,s}), \tag{4}$$

where $\mu_{h,s}$ and $\Sigma_{h,s}$ are the mean and covariance matrix of the color histogram similarity, which can be learned by using the EM algorithm on the labeled training data.

3.2.4. Joint Model of Identity and Appearance Similarity. By integrating the above statistical models of appearances and identity, the statistical model f in (1) can be updated as the joint distribution of appearance similarity and identity similarity, which are collectively denoted as $S = \{h_X - h_Y, s_X - s_Y, S_{\text{ID}}\}$:

$$\begin{aligned} &P_{\text{similarity}}(S \mid X(t), Y(t+\tau)) \\ &\quad = P_{\text{app}}(X(t), Y(t+\tau)) \cdot P_{\text{ID}}(X(t), Y(t+\tau)) \\ &\quad = P(h_X - h_Y, s_X - s_Y \mid X(t) = Y(t+\tau)) \\ &\qquad \cdot P\left(S_{\text{ID}}\left(\tilde{I}_Y\right) \mid X(t) = Y(t+\tau)\right). \end{aligned} \tag{5}$$

In (5), the joint distribution of appearance similarities and identity similarity is the product of the marginal distributions of each under the assumption that the appearance and identity are statistically independent. For each possible node pair, there is an associated multivariate mixture of Gaussians with unknown mean and variance, which can be estimated by using the EM algorithm. We can even relax the independence assumption provided that we have enough training samples to learn the covariance matrix of the joint distribution. Then, the cross-correlation function of departure and arrival sequences is updated as

$$R_{X,Y}(\tau) = \sum_{t=-\infty}^{\infty} P_{\text{similarity}}(S \mid X(t), Y(t+\tau)). \tag{6}$$

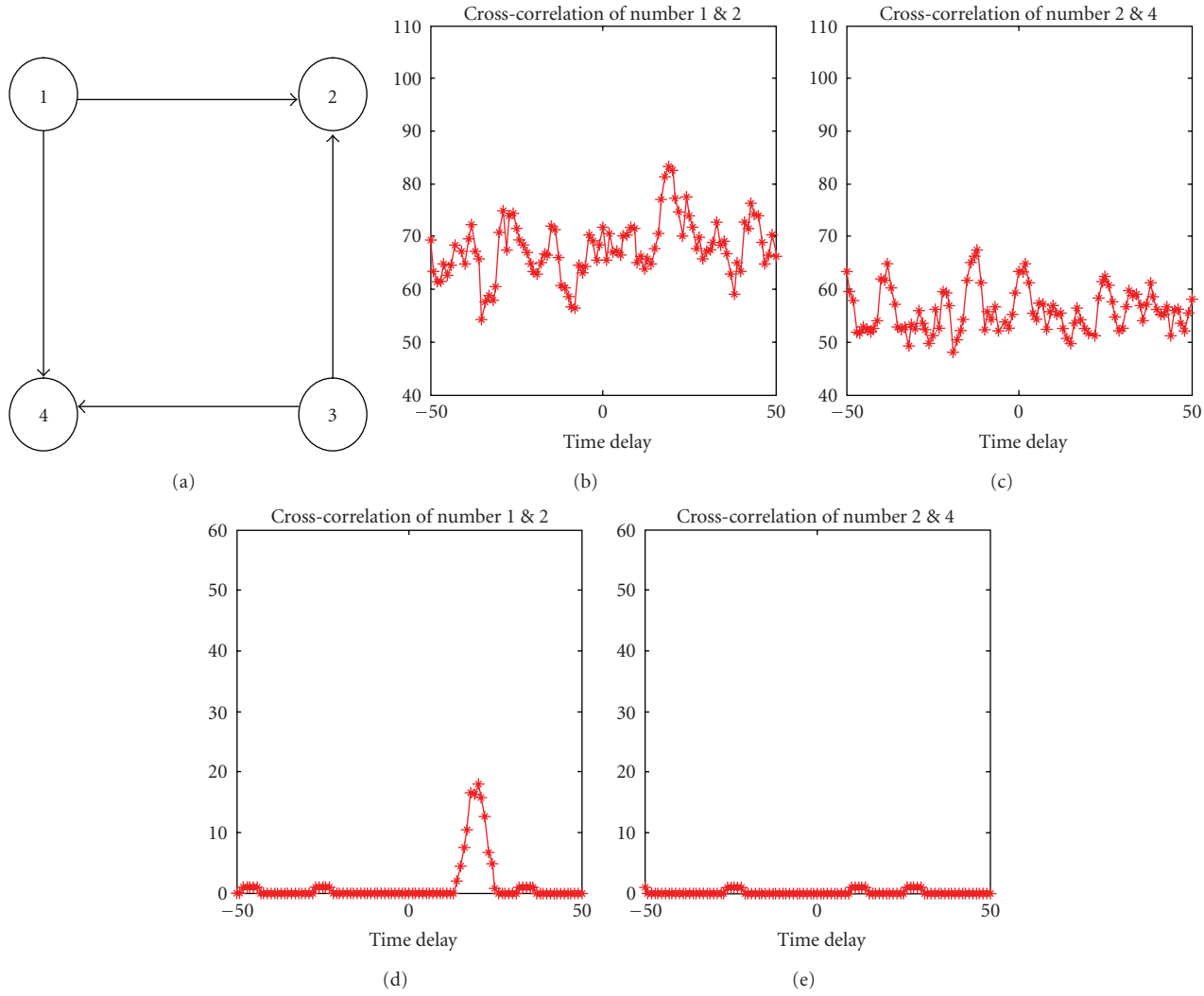

FIGURE 6: Example of a simple 4-node network for analysis. (a) The network topology. (b)–(e) The cross-correlations of node pairs 1-2, 2–4 of different approaches: (b), (c) are as in [15] and (d), (e) are our approach.

3.3. Network Topology Inference. We build a 4-node network (as shown in Figure 6(a)) to illustrate the importance of the identity in determining the network topology and the transition time between nodes. In the network, nodes 1 and 3 are departure nodes; 2 and 4 are the arrival nodes. The network is fully connected by the four links shown as arrows. The traffic data of 100 points is generated by a Poisson departure process *Poisson*(0.1), and the transition time follows the Gamma distribution *Gamma*(100, 5) as in [14]. The probability of the appearance similarity P_{app} is generated as a univariate Gaussian distribution $N(0, 1)$, and that of identity similarity P_{ID} from the mixture of Gaussians as in Figure 5.

The noisy cross-correlations by the previous approach in [5] (shown in Figures 6(b), and 6(c)) are replaced by the cleaner plots of our method (as in Figures 6(d), and 6(e)). Thus, the existence of possible links between different node pairs can be easier to infer from the cross-correlations with a loose threshold. Another possible advantage of our approach is that it can relieve the dependence on a large number of data samples for statistical estimation.

The mutual information (MI) between two temporal sequences ([5]) reveals the dependence between them:

$$I(X,Y) = \int p(X,Y)\log\frac{p(X,Y)}{p(X)\cdot p(Y)}dXdY = -\frac{1}{2}\log_2\left(1-\rho_{X,Y}^2\right), \tag{7}$$

where $\rho_{X,Y}^2 \approx \max(R_{X,Y}) - \text{median}(R_{X,Y})/(\sigma_x \cdot \sigma_y)$.

Thus, we can use the mutual information to validate the existence of the links identified in the network. As shown in the adjacency matrix in Figure 7(a), the links of "1 to 2", "1 to 4", "3 to 2", and "3 to 4" can be verified by the higher mutual information between them, which are shown as brighter grids.

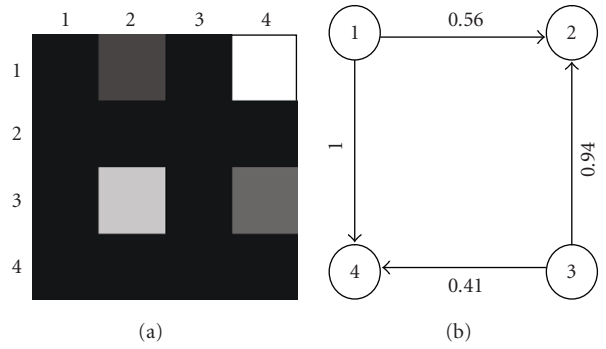

Figure 7: The network topology inference of the 4-node network: (a) the adjacency matrix of the mutual information between departure (row) and arrival (column) sequences; (b) the inferred weighted, directed graph of the connectivity.

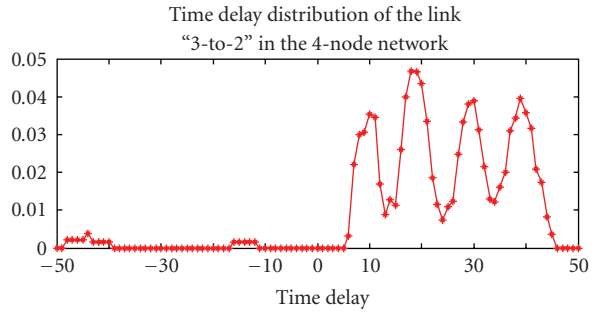

Figure 8: The multi-modal distribution of the time-delay τ.

The normalized mutual information is used as the weight of the links in the network topology graph (Figure 7 (b)):

$$W_{i,j} = \frac{I_{i,j}(X,Y)}{M_I}, \quad \text{in which } M_I = \arg\max_{(i,j)} \left(I_{i,j}(X,Y)\right). \tag{8}$$

3.3.1. Identifying Network Traffic Patterns. The traffic pattern over a particular link is characterized by the time-delay distribution, $P_{X,Y}(\tau)$, which can be estimated by normalizing the cross-correlation $R_{X,Y}(\tau)$:

$$P_{X,Y}(\tau) = \frac{R_{X,Y}(\tau)}{||R_{X,Y}(\tau)||}, \tag{9}$$

where $\|R_{X,Y}(\tau)\|$ is the area under the cross-correlation.

Depending on the moving object type, for example, pedestrians of different ages, mixture of pedestrians and vehicles, and so forth, the transition time distribution $P(\tau)$ has just a single mode (e.g., $T_0 = 20$ in Figure 6(d)), or multiple modes (e.g., 10, 20, 30 and 40 in Figure 8, resp.). The multi-modal transition time distribution in Figure 8 was obtained on the simulated 4-node network as in [14]. Specifically, the simulated distribution was generated by a mixture of Gamma distributions, that is, *Gamma*(100, 5), *Gamma*(25, 2.5), *Gamma*(225, 7.5), and *Gamma*(400, 10), to simulate the various speeds of objects.

3.4. Continuous Learning of Traffic Patterns and Network Topology. The learning algorithm described below operates at the lowest level, in the current implementation, where the bulk of work computation takes place. The same learning algorithm does not operate at different levels. At the camera level the results of entry/exit patterns form the association among cameras. In particular, the links between the entry/exit nodes from different cameras form the links between camera nodes. Similar association process is performed at the higher levels of the hierarchy.

The inferred traffic pattern (i.e., time delay distribution) is modeled as Gaussian Mixture Model (GMM) with parameters $\theta = (k, \alpha_m, \mu_m, \sigma_m^2)$ by using the Expectation-Maximization (EM) algorithm:

$$P_{X,Y}(\tau) = P_{X,Y}(\tau \mid \theta) \sim \sum_{m=1}^{k} \alpha_m \cdot N(\mu_m, \sigma_m^2). \tag{10}$$

In Figure 9, we show an example of GMM for modeling a single Gaussian of the time delay distribution. The statistics (i.e., the normalized occurrence as from (9)) of the time delays in the link "1 to 4" is shown in Figure 9(a), and its parameters are ($k = 1, \alpha_1 = 1, \mu_1 = 10, \sigma_1^2 = 4$), of which the Gaussian distribution is shown in Figure 9(b). The estimated GMM parameters by the EM algorithm are ($\widetilde{k} = 1, \widetilde{\alpha}_1 = 1, \widetilde{\mu}_1 = 9.956, \widetilde{\sigma}_1^2 = 4.247$) shown in Figure 9(c). We can find that the estimated GMM is capable to model the true traffic pattern. For the efficiency of the continuous learning system, a "change-detection" mechanism is employed to determine if the latent traffic pattern changes or not. The further time-consuming MCEM-based continuous learning is triggered only if a significant deviation of the current traffic pattern from the historical ones stored in the database is detected. After the continuous learning, the inferred GMMs of the traffic pattern are sent to update the traffic-pattern database. The overview of the continuous learning of traffic patterns and network topology is illustrated in Figure 10.

3.4.1. Traffic Pattern Change Detection. When the new data (departure/arrival sequences, the identities, etc.) for an established link ("$i \rightarrow j$") arrive at time t and the approximate correspondence between departures and arrivals is established by the recognized identities (I_X, I_Y), the time-delay distribution (i.e., traffic pattern $P_{X,Y}^t(\tau)$) at time t can be approximately inferred by the temporal correlation function as described in Sections 3.2 and 3.3. The current traffic pattern $P_{X,Y}^t(\tau)$ will be checked with the corresponding historical traffic pattern at day l (modeled as the GMM $\theta^{(l)}$) stored in the database by using the Kullback-Leibler divergence:

$$\begin{aligned} d\left(P_{X,Y}^t(\tau), \theta^{(l)}\right) &= D_{KL}(Q \,||\, P) \\ &= \int_{-\infty}^{\infty} Q(\tau) \log \frac{Q(\tau)}{P_{X,Y}^t(\tau)} d\tau, \end{aligned} \tag{11}$$

where

$$Q(\tau) = \text{GMM}\left(\tau \mid \theta^{(l)}\right) \sim \sum_{m=1}^{k} \alpha_m^{(l)} \cdot N\left(\mu_m^{(l)}, \sigma_m^{2(l)}\right). \tag{12}$$

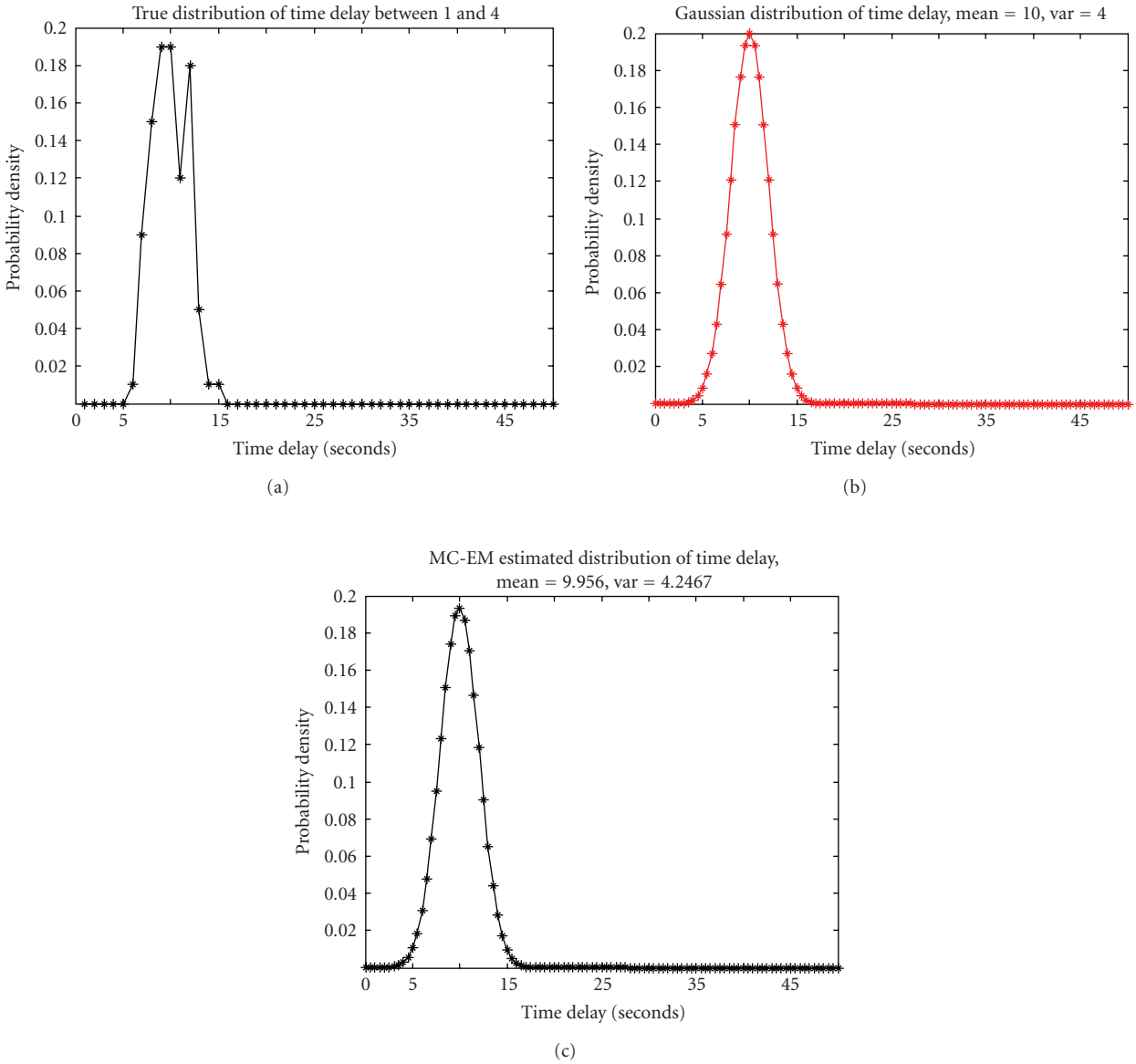

Figure 9: (a) The true distribution of time delay between nodes 1 and 4, (b) the GMM of the true time delay distribution, and (c) the estimated GMM of the time delay distribution by the EM method.

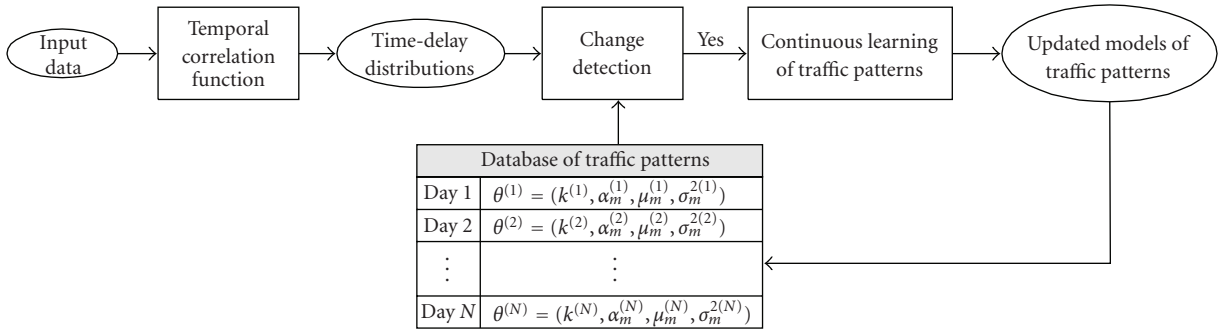

Figure 10: The overall approach for continuous learning.

The "distance" measure between the current and historical patterns will be evaluated with a predefined threshold ($\delta_{X,Y}$) to detect if there is a significant change.

The following MCEM-based continuous learning of traffic patterns and network topology is triggered after the detection of a significant change in the current traffic pattern.

3.4.2. MC-EM Algorithm. When the new input data (departure/arrival sequences, the identities, etc.) arrive at time t, the approximate correspondence between departures and arrivals can be established by the recognized identities (I_X, I_Y). However, there exist some false correspondences indicated by identities with low similarity scores. There may be ε of false identities assuming that the average accuracy of the face recognition system is (1- ε). We need to reestimate the correspondence for the ε of identities with the lowest similarity score probability to approach the true correspondence π. The reestimation is performed by an EM algorithm with the Markov Chain Monte Carlo (MCMC) technique.

The classical EM algorithm is not appropriate for this purpose since the enumeration of all possible correspondences is intractable. Therefore, the MC-EM algorithm is used for this task in which the MCMC sampling technique is used in the E step.

E-step. It calculates the expected log likelihood of the complete data given the current parameter guess:

$$\begin{aligned} Q\left(\theta, \theta^{(i-1)}\right) &= E\left[\log p(\tau \mid \theta) \mid X, Y, \pi, \theta^{(i-1)}\right] \\ &= \frac{1}{M} \sum_{m=1}^{M} \log p\left(\pi^{(m)}, X, Y \mid \theta\right). \end{aligned} \tag{13}$$

The expectation in (13) is intractable to enumerate all possible permutations. Therefore, we use the MCMC sampling to generate M samples for approximation as in (9). The parameter guess is initialized as the prior inferred parameters ($\theta^{(0)} = \theta^{t-1}$).

M-step. It updates our current parameter guess with a value that maximizes the expected log likelihood:

$$\begin{aligned} \theta^{(i)} &= \arg\max_{\theta} Q\left(\theta, \theta^{(i-1)}\right) \\ &= \arg\max_{\theta} \left[\frac{1}{M} \sum_{m=1}^{M} \log p\left(\pi^{(m)}, X, Y \mid \theta\right) \right]. \end{aligned} \tag{14}$$

We iterate over the E and M steps until we obtain a final estimate of θ. At each iteration of the algorithm, the likelihood increases and the process is terminated when subsequent iterations result in very small changes to θ. Algorithm 1 shows the pseudocode for the MCMC sampling.

LOOP:
1. sample $\pi^{(k+1)}$ from proposal;
2. sample U from an uniform distribution $U(0, 1)$;
3. $\alpha = \min\left(1, \dfrac{p(\pi^{(k+1)}, X, Y \mid \theta)}{p(\pi^{(k)}, X, Y \mid \theta)}\right)$;
4. if $U \leq \alpha$, then
   $\pi^{(k+1)}$ is accepted;
   else
   $\pi^{(k+1)}$ is rejected.
   end if
end LOOP

Algorithm 1: Markov Chain Monte Carlo Sampling Algorithm.

4. Experimental Results

We tested our proposed approach in simulation and in the real-life experiments, and compared it with the appearance-integrated approach [5], when applicable.

4.1. Simulated Experimental Results

4.1.1. Description of Network Simulation. The simulation is based on the "*entry/exit*" level of the multilayered network architecture illustrated in Figure 3. Since we focus on the connectivity inference in non-overlapping FOVs, the nodes with all connections within the same FOV are omitted. Thus, the simulated network has 18 departure/arrival nodes and 13 valid directed links. Some nodes, for example, node 11, work as both "departure" and "arrival." Some node pairs, for example, 6 and 22, have two unidirectional links, which models the asymmetric traffic patterns between the throughput nodes such as doors. The traffic data of 100 points are generated by a *Poisson*(0.1) departure process, and the transition time follows Gamma distributions, for example, *Gamma*(100, 5), *Gamma*(25, 2.5), and so forth. The probability of identity similarity P_{ID} is generated by a mixture of Gaussians as shown in Figure 4. For simplicity, the probability of appearance similarity is modeled by a univariate Gaussian $N(0, 1)$.

4.1.2. Learning Network Topology. The appearance and identity-based approach proposed in the paper is tested on the simulated traffic data. We assume that all the transition time distributions are single-mode. The cross-correlations with the appearance and identity (as in (6)) for twelve valid and twelve invalid links are shown in Figures 11(a) and 11(b), respectively. For comparison, Figures 10(c) and 10(d) show appearance-based cross-correlations [5] for the same valid and invalid links, respectively. It can be seen that our approach can highlight the peaks for the valid links and repress fluctuations for the invalid links, which greatly improves the peak signal-to-noise ratios of the estimation.

As to the link validation, we calculate the mutual information of departure and arrival sequences at various nodes and show the adjacency matrices in Figures 12(a) and 12(b). Based on the adjacency matrices, the topology graphs

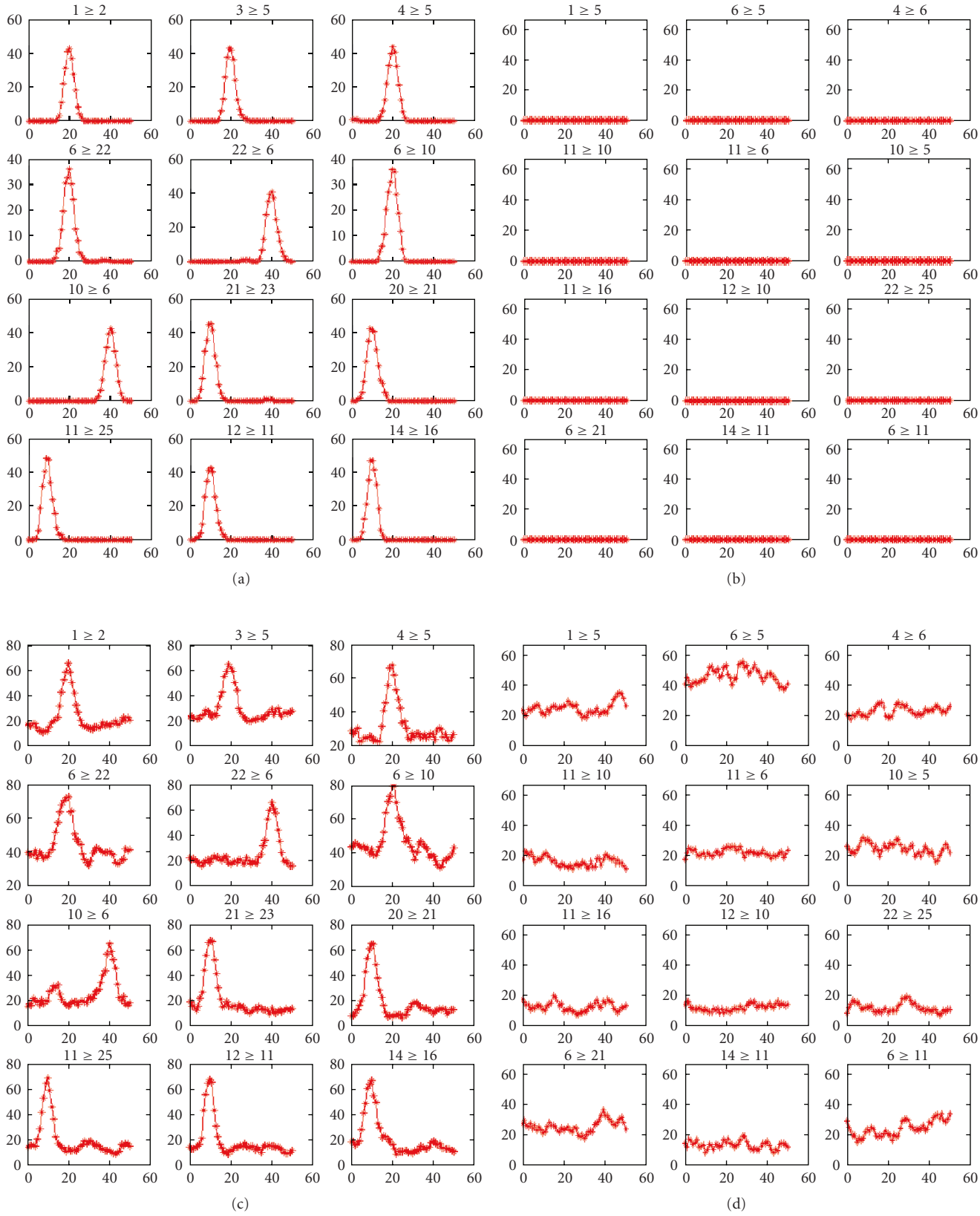

FIGURE 11: The estimated cross-correlations. (a), (b) our proposed approach, (c), (d) The previous approach in [5]. (a), (c) are for valid links and (b), (d) for invalid links.

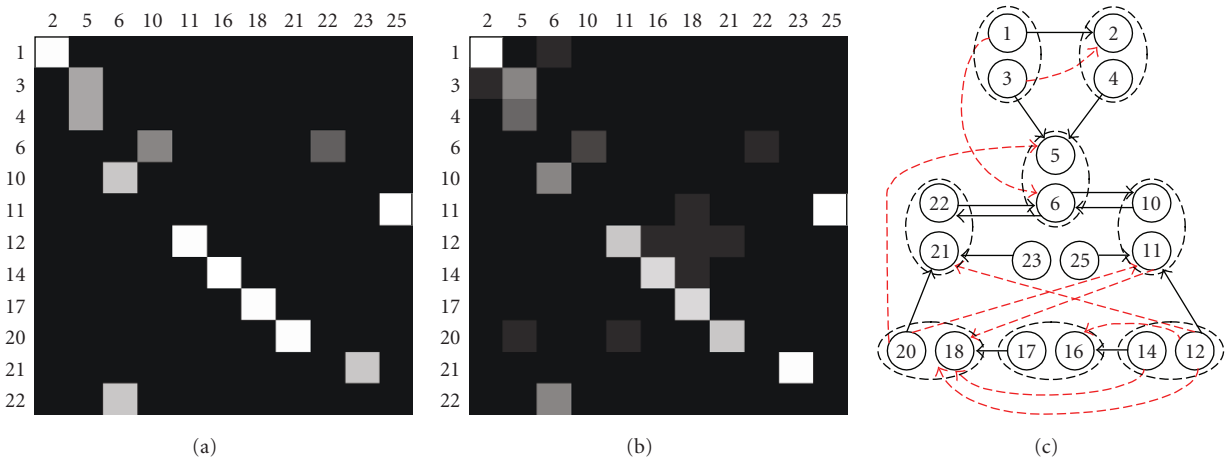

Figure 12: The adjacency matrices of mutual information: (a) by our approach; (b) by the previous approach in [5]; (c) the inferred topology graph. Those nine false links inferred by the adjacency matrix in (b) are marked as dashed links in (c).

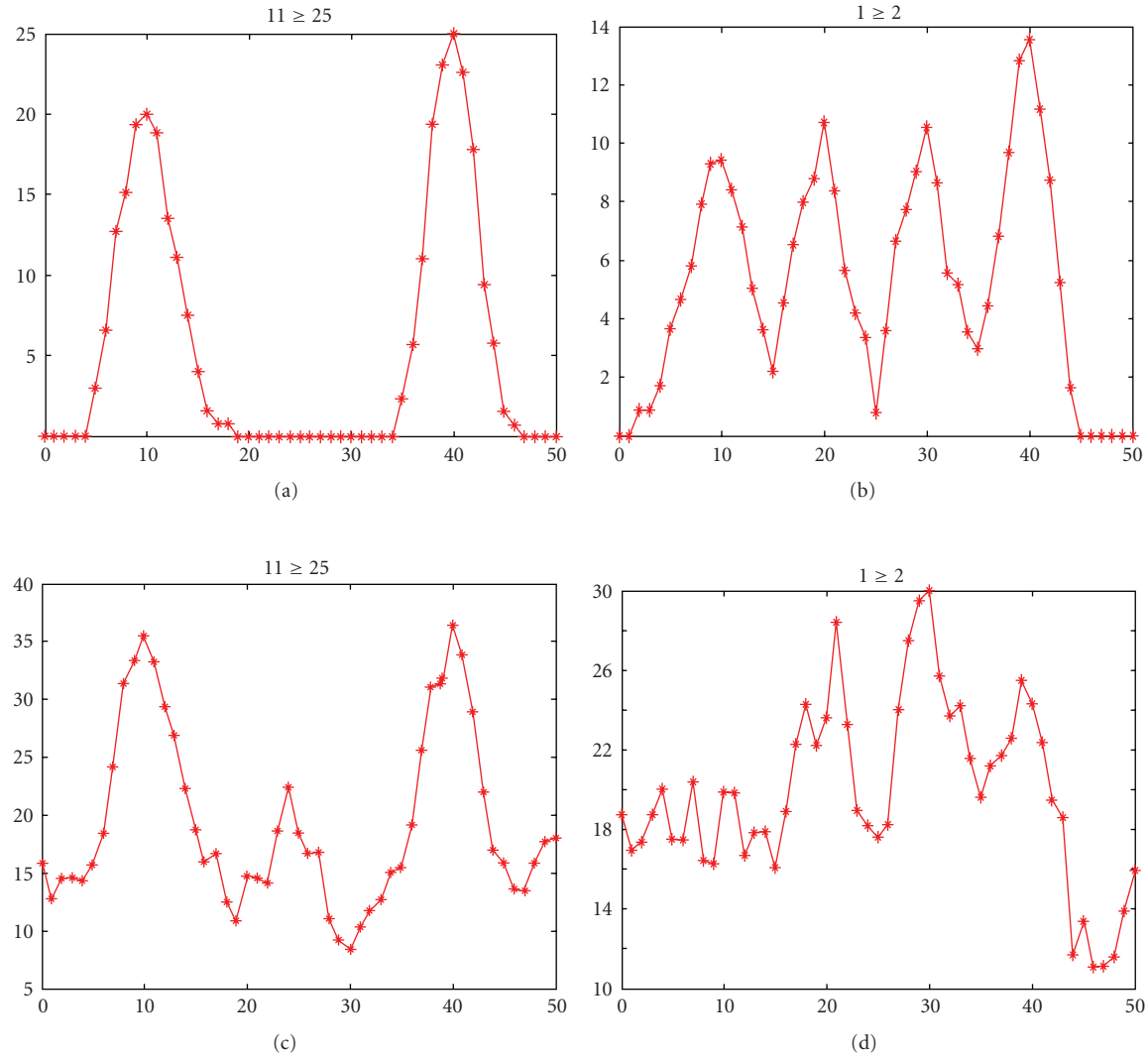

Figure 13: The estimated multi-modal traffic patterns by (a), (b) our approach; (c), (d) as in [5].

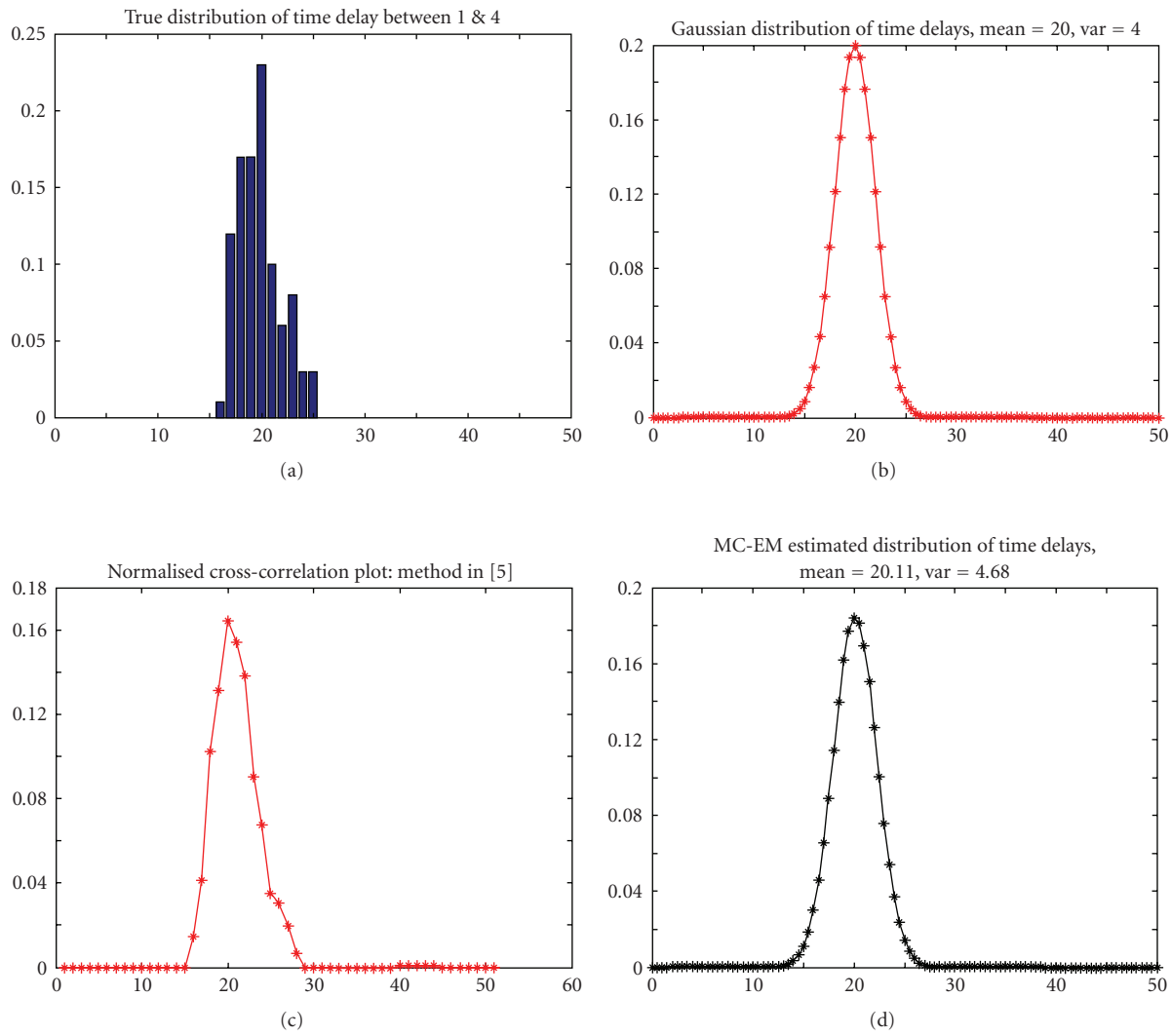

Figure 14: (a) The true distribution of time delay between nodes 1 and 4, (b) the Gaussian distribution of the true time delays, (c) the estimated time delay distribution by the cross-correlation method, and (d) the estimated time delay distribution (single Gaussian) by the MC-EM method.

are inferred as shown in Figure 12(c). In addition to the 13 valid links (shown as the solid lines), the appearance-based approach [5] also generates nine invalid links (dashed lines), which are mainly concentrated on the throughput nodes, for example, 6, 11, and 21.

4.1.3. Learning Multimodal Traffic Patterns. To illustrate the capability of learning multi-modal traffic patterns, we simulate the two-mode (e.g., at 10 seconds and 40 seconds) and four-mode (e.g., at 10 s, 20 s, 30 s and 40 s) transition time distributions by using the mixture of Gamma distributions as in Section 3.3.1 . The estimated cross-correlations are shown in Figure 13. Our approach (as in Figures 13(a), and 13(b)) correctly restores the two modes and four modes, while there are three outstanding peaks in Figure 13(c) and multiple peaks in Figure 13(d) by the appearance-based approach [5]. This result illustrates the capability of our approach to learn the multi-modal traffic patterns.

4.1.4. Example of Continuous Learning of Traffic Patterns in a Less Cluttered Scenario. First, we examine the continuous learning in a less cluttered scenario, for example, the hall way in a building on campus. The subjects in the traffic are mostly adults with very few disabled. Therefore, the traffic pattern usually shows a single peak centered at the most common transition time. The location of the peak (i.e., the common transition time) depends on the density of traffic: the more crowded the traffic is, the longer the common transition time will be taken. It is well known that the traffic in the school buildings varies a lot between the instruction time and the off-peak time. It constitutes a dynamically changing environment, which is ideal for the continuous learning.

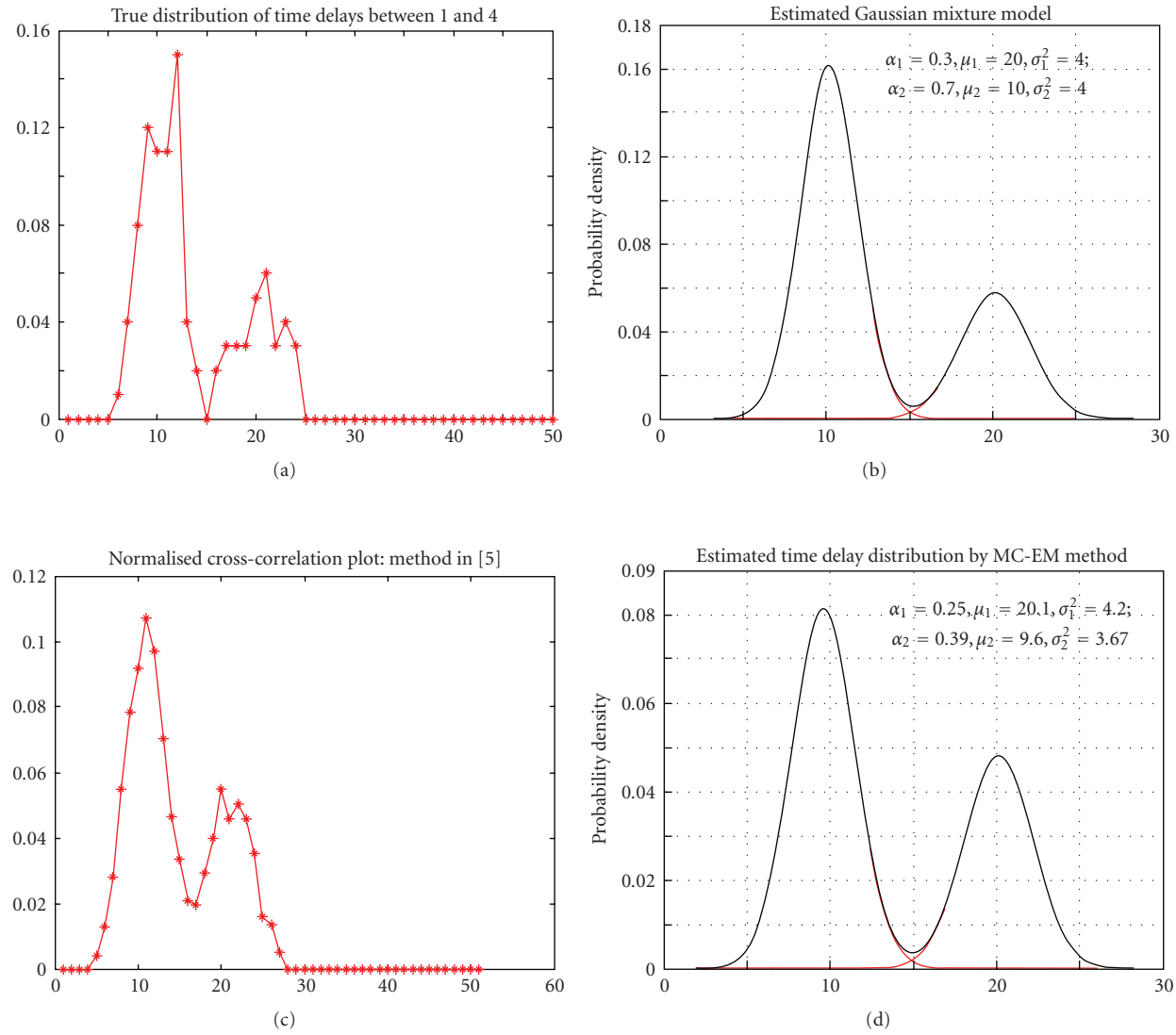

Figure 15: (a) The true distribution of time delay between nodes 1 and 4, (b) the Gaussian distribution of the true time delays, (c) the estimated time delay distribution by the cross-correlation method, and (d) the estimated time delay distribution (GMM) by the MC-EM method (at the fist iteration).

In this example, a significant change has been detected in the distribution of time delays at a later time, that is, $P^t_{X,Y}(\tau)$. It still contains a single Gaussian with the different means ($\mu = 20$) and the same variance ($\sigma^2 = 4$). The time delay distribution is continuously learned by the MC-EM algorithm with the initial guess as the previously estimated parameters (θ^{t-1}) and the comparison between the estimated traffic pattern estimated by the MC-EM and the cross-correlation methods is shown in Figure 14.

4.1.5. Example of Continuous Learning of Traffic Patterns in a Cluttered Scenario. Let us consider a more cluttered scenario, for example, a pedestrian path in a shopping center. The composition of the pedestrians varies during the business hours and the behaviors of the subjects may also change. In this example, we examine a special case when the distribution of time delays has changed from a single-peak to the multimodal distribution. The multimodality reflects significantly different patterns in the group of subjects, that is, the adults and the elderly, or the normal and the disabled. Therefore, the Gaussian Mixture Model (GMM) is used to model the multimodality. As shown in Figure 15, there are two Gaussian components in the GMM centered at 10 and 20, respectively.

Figure 16 shows the improved estimation of the multimodal time delay distribution at different iterations of the MC-EM algorithm.

4.1.6. Comparisons of Different Approaches for Topology Inference. For performance evaluation, we compare the following four approaches based on the number of correctly detected links in the camera network:

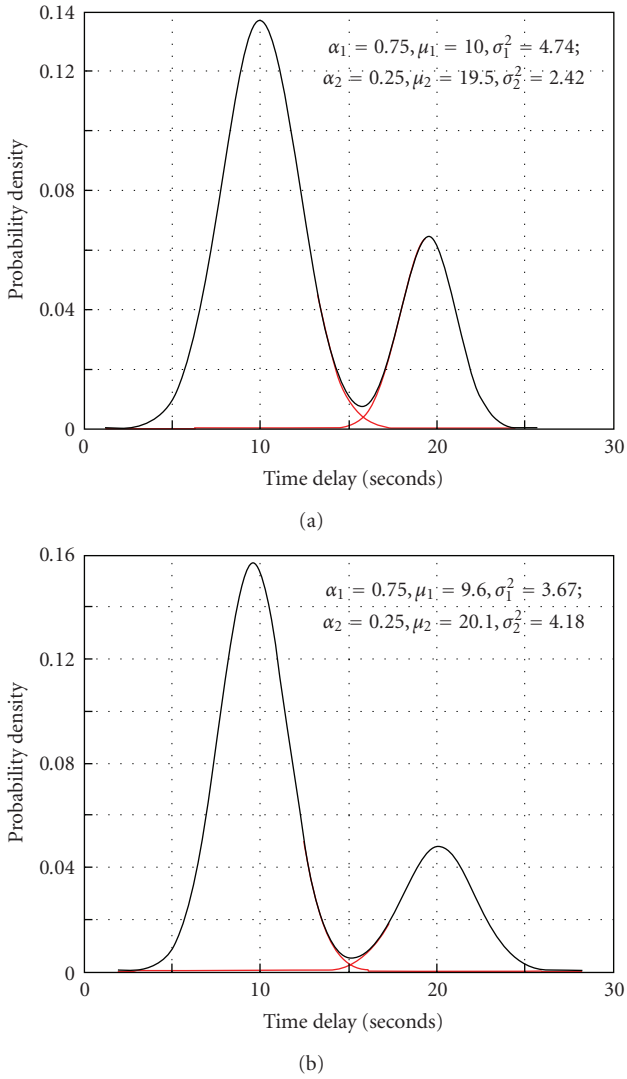

Figure 16: (a) The estimated time delay distribution by MC-EM at the 3rd iteration; (b) the estimated time delay distribution (GMM) by the MC-EM method at the 10th iteration.

(1) "static baseline": the appearance-integrated method in [5];

(2) "static CC": the appearance and identity-integrated cross-correlation method without continuous learning;

(3) "continuous baseline": the continuous learning method with only appearance considered (without identity);

(4) "proposed method": the continuous learning method as discussed in Section 3.4.

The experiment is conducted at four distinct times: time instance "1" as the initial time and others mean the moments when the network topology and traffic patterns significantly change. The performance accuracy is defined as the percentage of the correctly detected links versus the total number of valid links in the changed network, and

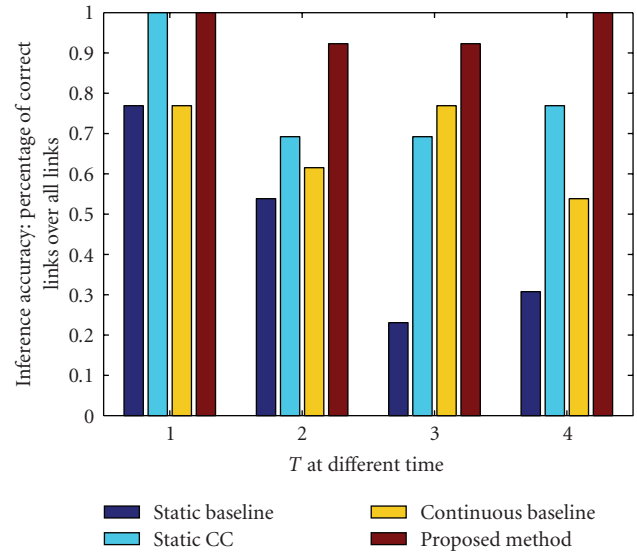

Figure 17: The performance comparison of the proposed method with other approaches: "static baseline," "static CC," and "continuous baseline."

the results are shown in Figure 17. From this figure, we can see that the proposed method achieves the best performance all the time. On the contrary, the methods without subject identity are much lower than our proposed method, and the performance of the "static CC" approach also deteriorates when the traffic environment changes.

4.2. Real-Life Experimental Results

4.2.1. Description of the Experimental Setup. The experimental setup of the distributed camera network is illustrated in Figure 18. As in the simulation, it follows the topology graph in Figure 3. Within it, there are nine cameras, in which six are on the tables (marked as circles) and three are on the ceiling (marked as triangles) distributed in two rooms on two different floors. There are four doors monitored by four cameras, where the heavy traffic occurs. There are also some barriers in the rooms that constrain possible paths.

We collected data on a test set of ten peoples: each person walked through the monitored environment ten times, totally 100 observations. The identification system is under construction so that we simulated the identity similarity distribution according to the mixture of Gaussians. After a manual selection of entry/exit points in each FOV (as colored ellipses in Figure 18), the object detection and tracking was employed to detect the departure and arrival events. Subsequently, the appearance similarity was calculated, and the probability of the appearance similarity was calculated on the estimated distribution P_{app} by using the EM algorithm and the labeled training data.

4.2.2. Learning Network Topology and Identifying Time-Varying Traffic Patterns. The proposed approach was tested on the real-life data to infer the network topology. It

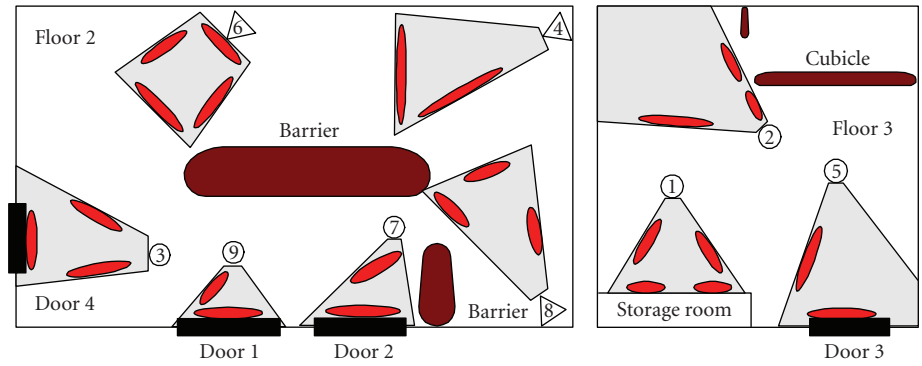

FIGURE 18: Experiment setup of the camera network showing the locations, FOVs, and entry/exit points of the cameras.

(a) (b) (c)

FIGURE 19: The example of false correspondence by appearance similarity metrics between different subjects. (a), (b) one subject observed at nodes 16 and 6, respectively; (c) the other one at node 4.

FIGURE 20: The observed departure and arrivals of four male or female objects in the FOVs of two cameras.

successfully recovered the topology of the camera network without any false link. However, the appearance-based approach [5] established several false links, to name a few, "4 to 6" and "4 to 16," by accumulating false correspondences. For example, in Figures 19(a) and 19(b) is the same subject, and Figure 19(c) is another one. Their identities (i.e., faces) are shown in the corners of each frame. Unfortunately, the false correspondences "a = c" and "b = c" are established by using the appearance similarity metrics; therefore, the false links "4 to 6" and "4 to 16" are inferred by accumulating these false correspondences.

It is challenging to learn the time-varying traffic patterns due to the unknown correspondence. Our appearance and identity-integrated approach provides a possible solution to this problem. As an example, for the scenes and observed traffic pattern as shown in Figure 20, the subjects in the traffic have both genders. In this case we find a two-mode pattern.

5. Conclusions

A multi-layered camera network architecture with nodes as entry/exit points, cameras, and clusters of cameras at different layers is proposed. Unlike existing methods that used discrete events or appearance information to infer the network topology at a single level, this paper integrates face recognition that provides robustness to appearance changes and better models the time-varying traffic patterns in the network. The statistical dependence between the nodes, indicating the connectivity and traffic patterns of the camera network, is represented by a weighted directed graph and transition times that may have multi-modal distributions. The traffic patterns and the network topology may be changing in the dynamic environment. We propose a Monte Carlo Expectation-Maximization algorithm-based continuous learning mechanism to capture the latent dynamically changing characteristics of the network topology. In the experiments, a nine-camera network with twenty-five nodes (at the lowest level) is analyzed both in simulation and in real-life experiments and compared with previous approaches.

For the applicability of our approach the face of the subjects should be visible at entry and exit points. Can this happen in realistic conditions? If the cameras are placed in corridors frontal face will be visible. For other situations in a camera network different cameras can be suitably placed for frontal face recognition in video [23, 24]. However, there will be situations where frontal face will not be visible at entry/exits. In those situations, side face (not frontal face) can be recognized in video [22]. In situations, when face recognition is not at all possible, the time delay-based approach will characterize the ultimate performance.

References

[1] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar, "Distributed localization of networked cameras," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 34–42, Nashville, Tenn, USA, September 2006.

[2] A. Rahimi, B. Dunagan, and T. Darrell, "Simultaneous calibration and tracking with a network of non-overlapping sensors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. 187–194, Washington, DC, USA, June-July 2004.

[3] R. Fisher, "Self-organization of randomly placed sensors," in *Proceedings of the 7th European Conference on Computer Vision (ECCV '02)*, vol. 2353, pp. 146–160, Copenhagen, Denmark, May 2002.

[4] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 205–210, Washington, DC, USA, June-July 2004.

[5] C. Niu and E. Grimson, "Recovering non-overlapping network topology using far-field vehicle tracking data," in *Proceedings of the International Conference on Pattern Recognition (ICPR '06)*, vol. 4, pp. 944–949, Hong Kong, August 2006.

[6] D. Marinakis, G. Dudek, and D. J. Fleet, "Learning sensor network topology through Monte Carlo expectation maximization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 4581–4587, Barcelona, Spain, April 2005.

[7] X. Zou, B. Bhanu, B. Song, and A. K. Roy-Chowdhury, "Determining topology in a distributed camera network," in *Proceedings of the International Conference on Image Processing (ICIP '07)*, vol. 5, pp. 133–136, San Antonio, Tex, USA, September 2007.

[8] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, Morgan Kaufmann, San Francisco, Calif, USA, 3rd edition, 2003.

[9] M. McCahill and C. Norris, "Analysing the employment of CCTV in European cities and assessing its social and political impacts," No. 3: CCTV in Britain, http://www.urbaneye.net.

[10] A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for sensor network self-calibration," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, 2005.

[11] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.

[12] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 35, no. 3, pp. 397–408, 2005.

[13] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '03)*, vol. 2, pp. 952–957, Nice, France, October 2003.

[14] K. Tieu, G. Dalley, and W. E. L. Grimson, "Inference of non-overlapping camera network topology by measuring statistical dependence," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1842–1849, Beijing, China, October 2005.

[15] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, Berlin, Germany, 2001.

[16] F. Dellaert, "Addressing the correspondence problem: a markov chain monte carlo approach," Tech. Rep., Carnegie Mellon University School of Computer Science, Pittsburgh, Pa, USA, 2000.

[17] G. Wei and M. Tanner, "A Monte-Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 699–704, 1990.

[18] X. Zou and B. Bhanu, "Anomalous activity classification in the distributed camera network," in *Proceedings of the International Conference on Image Processing (ICIP '08)*, pp. 781–784, San Diego, Calif, USA, October 2008.

[19] Y. Xu, A. Roy-Chowdhury, and K. Patel, "Pose and illumination invariant face recognition in video," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, Minneapolis, Minn, USA, June 2007.

[20] J. Yu, B. Bhanu, Y. Xu, and A. K. Roy-Chowdhury, "Super-resolved facial texture under changing pose and illumination," in *Proceedings of the International Conference on Image Processing (ICIP '07)*, vol. 3, pp. 553–556, San Antonio, Tex, USA, September 2007.

[21] J. Yu, *Super-resolution and facial expression for face recognition in video*, Ph.D. thesis, Department of Electrical Engineering, University of California at Riverside, Riverside, Calif, USA, 2007.

[22] X. Zhou and B. Bhanu, "Integrating face and gait for human recognition at a distance in video," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 37, no. 5, pp. 1119–1137, 2007.

[23] "Videoweb," http://www.vislab.ucr.edu/RESEARCH/Projects/VideoWeb/index.htm.

[24] H. Nguyen and B. Bhanu, "VideoWeb: design of a wireless camera network for real-time monitoring of activities," in *Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '09)*, Como, Italy, August-September 2009.

[25] C. Soto, B. Song, and A. Roy-Chowdhury, "Distributed multi-target tracking in a self-configuring camera network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, Miami, Fla, USA, June 2009.

[26] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[27] R. Wang and B. Bhanu, "Learning models for predicting recognition performance," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1613–1618, Beijing, China, October 2005.