Traffic Modeling and Prediction Using Sensor Networks: Who Will Go Where and When?

ZAIHONG SHUAI, University of California, Merced
SANGSEOK YOON and SONGHWAI OH, Seoul National University
MING-HSUAN YANG, University of California, Merced

We propose a probabilistic framework for modeling and predicting traffic patterns using information obtained from wireless sensor networks. For concreteness, we apply the proposed framework to a smart building application in which traffic patterns of humans are modeled and predicted through human detection and matching of their images taken from cameras at different locations. Experiments with more than 100,000 images of over 40 subjects demonstrate promising results in traffic pattern prediction using the proposed algorithm. The algorithm can also be applied to other applications, including surveillance, traffic monitoring, abnormality detection, and location-based services. In addition, the long-term deployment of the network can be used for security, energy conservation, and utilization improvement of smart buildings.

Categories and Subject Descriptors: C.2.0 [**Computer-Communication Networks**]: General; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms: Algorithms, Design, Experimentation, Performance

Additional Key Words and Phrases: Camera sensor network, traffic modeling and prediction, pedestrian detection, image matching, smart cameras

ACM Reference Format:
Shuai, Z., Yoon, S., Oh, S., and Yang, M.-H. 2012. Traffic modeling and prediction using sensor networks: Who will go where and when? ACM Trans. Sensor Netw. 9, 1, Article 6 (November 2012), 28 pages.
DOI = 10.1145/2379799.2379805 http://doi.acm.org/10.1145/2379799.2379805

1. INTRODUCTION

In this article, we propose a probabilistic framework for modeling traffic pattern of moving objects using information acquired from wireless sensor networks. The traffic patterns here refer to the moving trends of humans, vehicles, or other moving objects within the regions of interest. We assume that the ways in which objects move around follow some regular patterns based on preference and limitations of area layouts. Based on the image observations, we extract useful information to learn how the objects move in the scenes. For example, we can predict the transition probability of an object

Preliminary results of this work [Shuai et al. 2010] were presented in the *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*.
Z. Shuai is supported in part by the Army Research Office MURI grant W911NF-06-1-0076. S. Yoon and S. Oh are supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0027155). M.-H. Yang is supported in part by a Google Faculty Award.
Authors' addresses: Z. Shuai and M.-H. Yang, Electrical Engineering and Computer Science, University of California, Merced, 5200 N. Lake Road, Merced, CA 95344; email: {zshuai, mhyang}@ucmerced.edu; S. Yoon and S. Oh, Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea; email: {skywalker, songhwai}@snu.ac.kr.

moving from the sensing region of one sensor to another. In addition, we can estimate the expected traveling time for an object moving between sensing regions with the predicted transition probabilities.

In our formulation, no overlapping sensing regions are required, and the sensing region of each sensor can have different shapes. The sensors are not calibrated, that is, we do not know their accurate positions or viewpoints. This scenario entails an efficient and effective data association algorithm to match objects observed by different sensors, as there are multiple objects moving freely in the scenes. For concreteness, we describe our framework using a smart building application in which we show humans can be identified and matched based on their images acquired from cameras with different fields of view. The proposed framework can be applied, with different sensing devices, to numerous problems.

—*Abnormality detection*. To monitor areas and detect abnormal activities, we first can model the traffic patterns. With the learned traffic patterns, an abnormal event can be identified if the sensor network detects an unusual pattern significantly different from what is modeled. Such information can be useful and important for operators to respond.

—*Surveillance*. In public places such as shopping malls, airports, and parking lots, we can use the traffic patterns to infer entrances or exits that are likely to have high throughputs during some periods of time. Sufficient labor or physical resources can therefore be appropriated to handle potential congested traffic or emergency. On the other hand, mall exits with low throughput may be closed to reduce cost without affecting the influx or outflux of shoppers.

—*Location-based services*. When a moving object is detected by a sensor, such information can be passed to the sensors at its next possible stops. Consequently, with controlled focus and zoom, better images of that object can be obtained. Such traffic information is of critical importance for smart sensing rather than conventional passive sensing.

—*Energy conservation*. The long-time deployment of our system can also benefit energy conservation. With the learned traffic patterns, we have a global view of traffic flow with occupancy information, which is useful for controlling air conditioning, lighting, and other appliances in the regions of interest.

We conduct experiments in a smart building with a low-power, low-bandwidth distributed camera sensor network. With five CITRIC [Chen et al. 2008] camera motes placed at the intersections of stairways, hallways, and elevators, we show that traffic patterns of dwellers can be modeled and predicted well with the proposed model.

Based on our previous work [Shuai et al. 2010], we have made the following additional work. First, we detail the description and analysis of the proposed framework. Second, we conduct extensive experiments to validate the proposed framework with a dataset of over 100,000 images, compared to a dataset around 28,000 images in the previous work [Shuai et al. 2010]. Third, we give in-depth analysis and discussion on the experimental results in Sections 4.4 and 4.5. We also add preliminary results on human detection using the histogram of oriented gradients (HOG) [Dalal and Triggs 2005] running on CITRIC motes.

The contributions of our work are summarized as follows. We propose a probabilistic framework, based on a semi-Markov process, for modeling the traffic patterns. The proposed approach deals with identity uncertainty, and hence it is applicable for realistic situations in which a large number of objects move among the regions of interest and their identities are not known a priori. Due to intrinsic characteristics of cameras, the data association problem with visual data is challenging, as images are acquired under different viewpoints or lighting conditions. We present a maximum-likelihood estimation algorithm to address the data association problem from noisy image data. Furthermore, the proposed framework exploits both spatial and temporal information such that only local information between neighboring sensors is used, and thus, the computational load can be reduced. To validate the proposed model, we conduct large-scale experiments in which over 100,000 image frames were collected. Experimental results and analysis demonstrate the merits of the proposed algorithm.

The remainder of this article is organized as follows. We review the most relevant works on camera sensor networks and traffic pattern models in Section 2. We detail the problem formulation in Section 3 and put this work in the appropriate context. Numerous experiments have been carried out to validate the proposed algorithm in a smart building application. Experimental results and discussions are presented in Section 4. We conclude this article with remarks in Section 5.

2. RELATED WORK

There is a rich literature on sensor networks [Akyildiz et al. 2002; Soro and Heinzelman 2009], and a comprehensive review is beyond the scope of this work. In this section, we discuss the most relevant works in camera sensor networks and their applications for modeling human activities.

There has been a growing interest in applications with smart cameras, including tracking objects using multiple cameras [Pasula et al. 1999; Haritaoglu et al. 2000; Javed et al. 2003, 2005; Gilbert and Bowden 2006; Song and Roy-Chowdhury 2008], object identification [Huang and Russell 1997], people counting [Yang et al. 2003], and learning network topology [Niu and Grimson 2006]. Huang and Russell [1997] present a traffic monitoring system in which image matching and known traveling time are combined to establish vehicle correspondence between deployed camera sensors on a highway. However, it only models one single traffic pattern where the traffic generally follows highway lanes. Kettnaker and Zabih [1999] introduce a Bayesian formalization to reconstruct the paths of objects across multiple cameras. While the cameras have non-overlapping fields of view (FOV), they need to be calibrated so that object movements can be inferred. In addition, their system requires a predefined set of allowable paths, transition probabilities, and expected duration as prior information. Consequently, the proposed method has limited application domains. A method that exploits space-time cues (e.g., location of exits and entrances, moving directions, average traveling time, and object appearance) to establish object correspondences is presented in Javed et al. [2003]. Although the results are promising, the proposed method does not predict the traveling time of moving objects.

To track people moving across cameras, a method based on a stochastic transition matrix is developed [Dick and Brooks 2004] in which both a Kalman filter and Markov model are used. The Kalman filter is used to model short tracks between frames, whereas the Markov model is applied to cope with discontinuity and fast motion or movement that the Kalman filter cannot predict. However, this method relies on background subtraction, which is known to be problematic for long-term deployment. In addition, it does not model the traveling time of moving people. Spatial and visual cues are used in Javed et al. [2005] for tracking objects in multiple non-overlapping cameras. The non-parametric Parzen kernel function is used to estimate the space-time probability density function between each pair of cameras, thereby facilitating tracking with non-overlapping views. A method proposed in Gilbert and Bowden [2006] incrementally updates a transition matrix and color calibration mappings for tracking people across disjoint camera views. Song and Roy-Chowdhury [2008] propose a stochastic, adaptive strategy for tracking multiple people in non-overlapping camera networks. With its long-term feature dependency models, their system is able to determine feature correspondence and correct association errors. However, they assume that the distribution of the travel time between two nodes is known and people can always be tracked within the view of each camera, which is known to be a difficult problem. Makris et al. [2004] derive a statistical model to learn the topography of camera networks and can be extended to estimate the re-appearance location and time probabilistically. However, the learning algorithm of this work does not determine the correspondence of observed objects.

In this work, a camera sensor network is formed using the CITRIC camera motes [Chen et al. 2008]. There are numerous camera sensor platforms [Rahimi et al. 2005; Kulkarni et al. 2005; Feng et al. 2005; Downes et al. 2006; Kleihorst et al. 2007] which vary in configuration, processing capability, memory, and image resolutions. Unlike other sensor platforms, the processing unit is decoupled into two parts in a CITRIC mote—one for communication and the other for image processing, which differentiates it from other camera sensor motes. However, our framework can be implemented using any camera sensor platform other than the CITRIC mote.

In the last few years, there has been a rapidly growing interest in platforms and applications of camera sensor networks [Sundarraj et al. 2006; Yan et al. 2008; Ko et al. 2010; Keshavarz et al. 2006; Heath and Guibas 2007; Diaz et al. 2007; Kamthe et al. 2009; Lobaton et al. 2009]. The subjects of these applications vary from image matching, object tracking to distributed image searching, and object position estimation. Notwithstanding the demonstrated success in these applications, considerably few efforts have been made for developing efficient and effective data association algorithms to model the traffic patterns of moving objects.

What distinguishes our work from prior art is summarized as follows. First, it is not necessary for our algorithm to track objects or to reconstruct their whole paths in image sequences for analyzing traffic patterns. Instead, the proposed algorithm entails only local motion patterns of objects to be estimated. Second, our method is able to model the traveling time of moving objects. Furthermore, while the focus of our work is not on object tracking in the sensor network, the moving paths of these objects can be estimated probabilistically. In addition, the number of moving objects in the regions of interest can be inferred. With each mote reporting the number of objects entering/leaving the states (based on human detection) at any duration, a global view of the traffic flow within the camera sensor network can be constructed. Last but not least, we can also learn the topology with our framework when no spatial constraint is applied and no topology is known a prior.

3. PROBLEM FORMULATION

We present the proposed framework for modeling and predicting traffic patterns in this section. Our framework is generic and applicable to numerous problems, as we do not assume specific sensing region or topology of sensors. It can be applied to other sensor networks with different sensing devices (e.g., infrared, motion, and image sensors). For concreteness, we present the proposed framework with an application where traffic patterns of humans are modeled and predicted via images acquired from a camera sensor network.

3.1. Sensor Placement

Assume that there are N sensors in the network; we denote $\mathcal{R}$ as the entire region of interest which covers sensing areas of all the sensors, that is, $\{R_1, \dots, R_N\} \subset \mathcal{R}$, where R_n is the sensing region of sensor n. These sensing regions may be overlapped or not. Note that there exist regions uncovered by the sensors, so the union of $R_1, \dots, R_N$ is a subset of $\mathcal{R}$. These sensing regions are not assumed to have any particular structure in our formulation. As shown in Figure 1(a), there are five regions ($N = 5$) in this sensor network where R_3 and R_4 are overlapped.

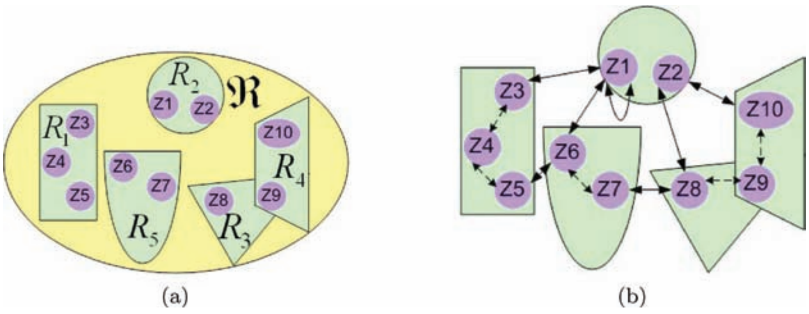


Fig. 1. Sensing regions and corresponding activity graph. (a) The sensing region of each sensor may be overlapped or not. They can have different shapes. (b) An activity graph. Z_1, Z_2 and Z_6, Z_7 are possible entrance/exit points in sensing region R_2 and R_5, respectively, and there is a path ($Z_1 \leftrightarrow Z_6$) between them. The graph representation distinguishes the U-turn and through traffic. The possible paths within and between R_n are represented by dotted and solid lines, respectively. For presentation clarity, not all possible paths are shown.

The possible entry/exit points within the whole sensing region are represented by S states $Z_1, \ldots, Z_S$ ($S \geq N$), as shown in Figure 1(b), where the states are denoted by circles. In this example, each sensing region is modeled by one or more states. On the other hand, a state may be covered by more than one sensor, for example, Z_9. The state index is unique regardless of which sensor it belongs to. Assume that a set of states, S_i, is covered by sensor i, then $\sum_{i=1}^{N} S_i = S + L$, where L is the total number of states that are covered by more than one sensor. In Figure 1, there are ten ($S = 10$) states, and one ($L = 1$) of them is covered by more than one sensors. With this formulation, the traffic patterns of interest refer to how objects travel from one state to another. We only consider the traffic patterns between R_n's.

The activity graph describes how objects move among the regions R_n. As shown in Figure 1(b), each vertex represents a state (i.e., a possible entrance or exit point of objects within that region), and each edge in the activity graph describes the possible path the objects can take between states. The activity graph differentiates traffic patterns, such as U-turn ($Z_1 \leftrightarrow Z_1$) and through traffic ($Z_1 \leftrightarrow Z_6$). Note that the activity graph accounts for all possible movements of all objects rather than the movements of one particular object.

3.2. Mobility Model and Observation Model

We model the traffic patterns of moving objects using a semi-Markov chain over the activity graph. Let X_k be the state of an object at time t_k. The state transition is modeled by

$$P(X_k = j | X_{k-1} = i) = p_{ij}, \tag{1}$$

where i and j denote states Z_i and Z_j, respectively. Unlike the conventional Markov chain, where the state transition happens instantaneously, we assume that there is a delay at each transition. Let T_k be the traveling time between X_{k-1} and X_k which is modeled by the exponential distribution with the following probability density function.

$$f(T_k = t | X_{k-1} = i, X_k = j) = \lambda_{ij} \exp(-\lambda_{ij} t). \tag{2}$$

With this semi-Markov chain model, there is no restriction on the amount of time an object stays in the same state. While the instantaneous transition between states of the conventional Markov chain is unrealistic, the traveling time durations are taken into account in this model. The initial state distribution is defined in a way similar to the

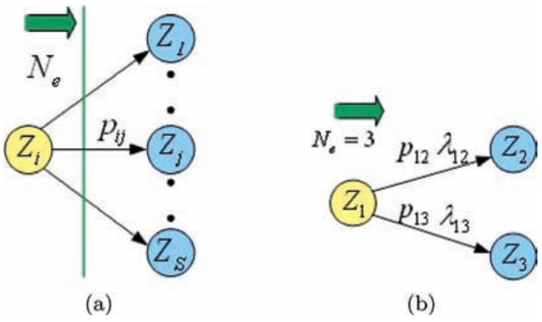


Fig. 2. Likelihood of the out-going transitions from state Z_i. (a) Assume that there are a total of N_e objects leaving Z_i. Their possible next stop is $Z_1, \ldots, Z_S$, and the corresponding transition probability is p_{ij}. (b) In this example, we assume that $N_e = 3$, that is, three objects are leaving from Z_1 and the possible next states are Z_2 and Z_3.

conventional Markov chain, and the semi-Markov chain describes the traffic patterns in the activity graph.

If an object is in R_n from time t_{k-1} to t_k, its properties, such as color, shape, texture, and other features, are observed by sensor n. The observation is modeled by the following function.

$$Y_t^n = h_n(\chi_t) + v^n, \tag{3}$$

where Y_t^n is the observation at time t by sensor n, h_n is the observation function which maps the intrinsic state χ of the object to observation Y for sensor n, and v^n accounts for noise.

In the activity graph, the traveling time on each edge can be measured by the state entry and exit time stamps. In our camera sensor network system, both images and traveling time durations are used as observations.

3.3. Learning With Known Identities

There are two sets of parameters to be estimated in order to model the traffic patterns using our semi-Markov chain model. They are state transition probabilities $\{p_{ij}\}$ and traveling time rates $\{\lambda_{ij}\}$.[1] In Section 3.2, we presented a model with the assumption that there is a single object. As there is no uncertainty about object identity, the parameters can be easily estimated. However, in a general setup, we need to consider the case with a large number of moving objects. Thus, the parameter estimation cannot be carried out unless the identities of these objects are resolved. In next two sections, we describe how we resolve the identity uncertainty (i.e., data association) problem and robustly estimate the parameters of the semi-Markov chain. In this section, we first assume that the identities are known.

For clarity of exposition, we present the method for estimating parameters for transition from a single state Z_i. The parameters associated with other states can be estimated in a similar manner. As shown in Figure 2(a), suppose that there are N_e objects that exit state Z_i, and the possible next states are $\{Z_j\}_{j=1,\ldots,S}$ with a corresponding transition probability p_{ij}. Then we can compute the likelihood of the out-going

[1]We also need to estimate the initial state distribution, but it is not discussed here since its estimation is trivial.

transitions from state Z_i as

$$\prod_{k=1}^{N_e}\prod_{j=1}^{S} p_{ij}^{\gamma_{kj}}, \tag{4}$$

where $\gamma_{kj} = 1$ if the object k exiting Z_i at time t is the same object that arrives at Z_j for the first time after t and $\gamma_{kj} = 0$, otherwise. If no object arrives at Z_j after time t, we also have $\gamma_{kj} = 0$.

Figure 2(b) shows one simple example to explain Equation (4). Here, we set N_e, the total number of objects leaving state Z_1, to 3. Among all three objects, the first two of them go to state Z_2 and another goes to Z_3. Intuitively, the likelihood of the out-going transition probability from Z_1 is $p_{12}p_{12}p_{13}$. As object 1 and 2 go to Z_2, $\gamma_{12} = 1$, $\gamma_{22} = 1$. Likewise, $\gamma_{33} = 1$ as object 3 goes to Z_3. Other γ's are all 0. From Equation (4), the likelihood is computed as

$$\begin{aligned}\prod_{k=1}^{N_e}\prod_{j=1}^{S} p_{ij}^{\gamma_{kj}} &= \prod_{k=1}^{3}\prod_{j=2}^{3} p_{ij}^{\gamma_{kj}}\\ &= \prod_{j=2}^{3} p_{1j}^{\gamma_{1j}} \cdot \prod_{j=2}^{3} p_{1j}^{\gamma_{2j}} \cdot \prod_{j=2}^{3} p_{1j}^{\gamma_{3j}}\\ &= p_{12}^{\gamma_{12}} \cdot p_{13}^{\gamma_{13}} \cdot p_{12}^{\gamma_{22}} \cdot p_{13}^{\gamma_{23}} \cdot p_{12}^{\gamma_{32}} \cdot p_{13}^{\gamma_{33}}\\ &= p_{12} \cdot p_{12} \cdot p_{13},\end{aligned}$$

which is the same as what we expect.

Once we know object identities, that is, γ's, we can estimate the maximum likelihood of the transition probabilities by solving a constrained optimization problem as follows.

$$\max_{p_{ij}} \prod_{k=1}^{N_e}\prod_{j=1}^{S} p_{ij}^{\gamma_{kj}} \quad \text{s.t.} \sum_{j=1}^{S} p_{ij} = 1, \tag{5}$$

which is equivalent to maximizing the Lagrange function $L(p_{ij})$ (as log is a monotonic increasing function),

$$\begin{aligned}L(p_{ij}) &= \log\left(\prod_{k=1}^{N_e}\prod_{j=1}^{S} p_{ij}^{\gamma_{kj}}\right) - \nu\left(\sum_{j=1}^{S} p_{ij} - 1\right)\\ &= \sum_{k=1}^{N_e}\sum_{j=1}^{S} \gamma_{kj}\log(p_{ij}) - \nu\left(\sum_{j=1}^{S} p_{ij} - 1\right),\end{aligned} \tag{6}$$

where ν is the Lagrange multiplier. For $j = 1, \ldots, S$, we set $\frac{\partial L(p_{ij})}{\partial p_{ij}} = 0$ and get

$$\hat{p}_{ij} = \frac{\sum_{k=1}^{N_e}\gamma_{kj}}{\nu}. \tag{7}$$

By using the constraint $\sum_{j=1}^{S} p_{ij} = 1$, we have

$$\hat{p}_{ij} = \frac{\sum_{k=1}^{N_e}\gamma_{kj}}{\sum_{k=1}^{N_e}\sum_{j=1}^{S}\gamma_{kj}}. \tag{8}$$

The traveling time rates can be solved similarly. The likelihood of traveling times from state Z_i is

$$\prod_{k=1}^{N_e}\prod_{j=1}^{S}(\lambda_{ij}\exp(-\lambda_{ij}t_{ij}))^{\gamma_{kj}}, \tag{9}$$

where t_{ij} is the traveling time when $\gamma_{kj} = 1$.

We use the same example in Figure 2(b) to explain Equation (9). With $N_e = 3$ and $S = 2$, we have

$$\begin{aligned}\prod_{k=1}^{N_e}\prod_{j=1}^{S}(\lambda_{ij}\exp(-\lambda_{ij}t_{ij}))^{\gamma_{kj}} &= \prod_{k=1}^{3}\prod_{j=2}^{3}(\lambda_{ij}\exp(-\lambda_{ij}t_{ij}))^{\gamma_{kj}}\\ &= \prod_{j=2}^{3}(\lambda_{1j}\exp(-\lambda_{1j}t_{1j}))^{\gamma_{1j}}\cdot\prod_{j=2}^{3}(\lambda_{1j}\exp(-\lambda_{1j}t_{1j}))^{\gamma_{2j}}\\ &\quad\cdot\prod_{j=2}^{3}(\lambda_{1j}\exp(-\lambda_{1j}t_{1j}))^{\gamma_{3j}}\\ &= \lambda_{12}\exp(-\lambda_{12}t_{12})\cdot\lambda_{12}\exp(-\lambda_{12}t_{12})\cdot\lambda_{13}\exp(-\lambda_{13}t_{13}),\end{aligned}$$

which is consistent with their transition probabilities.

The optimization problem is formulated as

$$\max_{\lambda_{ij}} L(\lambda_{ij}), \tag{10}$$

where

$$\begin{aligned}L(\lambda_{ij}) &= \log\left(\prod_{k=1}^{N_e}\prod_{j=1}^{S}(\lambda_{ij}\exp(-\lambda_{ij}t_{ij}))^{\gamma_{kj}}\right)\\ &= \sum_{k=1}^{N_e}\sum_{j=1}^{S}\gamma_{kj}\log(\lambda_{ij}\exp(-\lambda_{ij}t_{ij})).\end{aligned} \tag{11}$$

By solving

$$\frac{\partial L(\lambda_{ij})}{\partial\lambda_{ij}} = 0, \tag{12}$$

we have

$$\sum_{k=1}^{N_e}\sum_{j=1}^{S}\frac{\gamma_{kj}(1-\lambda_{ij}t_{ij})\exp(-\lambda_{ij}t_{ij})}{\lambda_{ij}\exp(-\lambda_{ij}t_{ij})} = 0, \tag{13}$$

and the maximum likelihood estimate of the traveling time rate is

$$\frac{1}{\hat{\lambda}_{ij}} = \frac{\sum_{k=1}^{S}\gamma_{kj}t_{ij}}{\sum_{k=1}^{S}\gamma_{kj}}. \tag{14}$$

The maximum likelihood estimation of $\hat{p}_{ij}$ and $\hat{\lambda}_{ij}$ in Equations (8) and (14) are intuitively correct. In Equation (8), the numerator $\sum_{k=1}^{N_e}\gamma_{kj}$ accounts for the number of objects traveling from Z_i to Z_j, while the denominator $\sum_{k=1}^{N_e}\sum_{j=1}^{S}\gamma_{kj}$ is the number of all objects leaving Z_i. The transition probability $\hat{p}_{ij}$ is then obtained by counting the frequency of objects entering state Z_j among all the objects leaving state Z_i. In the example shown in Figure 2(b), the estimated transition probability from state Z_1 to Z_2 is computed (using Equation (8)) as

$$\begin{aligned}\hat{p}_{12} &= \frac{\sum_{k=1}^{3} \gamma_{k2}}{\sum_{k=1}^{3}\sum_{j=2}^{3} \gamma_{kj}}\\ &= \frac{\gamma_{12}+\gamma_{22}+\gamma_{32}}{3}\\ &= \frac{2}{3}.\end{aligned}$$

As we assume that among all three objects, two of them go to Z_2, the estimated $\hat{p}_{12}$ reflects the frequency how objects move between these two states.

Similarly, in Equation (14), the average traveling time from Z_i to Z_j, that is, the reciprocal of traveling time rate $\hat{\lambda}_{ij}$, is estimated as the average time it takes for all the objects traveling from Z_i to Z_j. Using the same example just described, we assume it takes object 1 eight seconds and object 2 ten second to travel from Z_1 to Z_2. In addition, object 3 travels from Z_1 to Z_3 in 20 seconds. Using Equation (14), we have the estimation of average traveling time from Z_1 to Z_2 as

$$\begin{aligned}\frac{1}{\hat{\lambda}_{12}} &= \frac{\sum_{k=1}^{3} \gamma_{k2} t_{12}}{\sum_{k=1}^{3} \gamma_{k2}}\\ &= \frac{\gamma_{12}t_{12}+\gamma_{22}t_{12}+\gamma_{32}t_{12}}{\gamma_{12}+\gamma_{22}+\gamma_{32}}\\ &= \frac{8+10+0}{2}\\ &= 9.\end{aligned}$$

Likewise,

$$\begin{aligned}\frac{1}{\hat{\lambda}_{13}} &= \frac{\sum_{k=1}^{3} \gamma_{k3} t_{13}}{\sum_{k=1}^{3} \gamma_{k3}}\\ &= \frac{\gamma_{13}t_{13}+\gamma_{23}t_{13}+\gamma_{33}t_{13}}{\gamma_{13}+\gamma_{23}+\gamma_{33}}\\ &= \frac{0+0+20}{1}\\ &= 20.\end{aligned}$$

It is clear that the estimation of $\frac{1}{\hat{\lambda}_{12}}$ is exactly the average time it takes for two objects to travel from Z_1 to Z_2. As only one object travels from Z_1 to Z_3, the estimation of $\frac{1}{\hat{\lambda}_{13}}$ is the same as the traveling time of object 3.

However, in general, we do not have the identity information, and γ_{kj} are random variables. Hence, we cannot directly solve for the maximum likelihood estimates as just stated. To address this problem, we need first to resolve the identity uncertainty problem which is described in the next section.

3.4. Data Association

Using the observations from each sensor as input, the data association problem is to compute the matching probability of any two observations. While it is impossible to achieve an accurate hard decision about the identity of each object, the matching probability serves as a good candidate for soft decision.

Let m be an object entering state Z_i detected by sensor n, and let $\boldsymbol{Y}_k^{n,m} = \{\boldsymbol{Y}_t^{n,m} : t_{k-1} \leq t \leq t_k\}$ be the collection of measurements from the time the object m enters R_n (at time t_{k-1}) to the time the object exits (at time t_k). Without loss of generality, we assume that $\boldsymbol{Y}_k^{n,m}$ is a series of color histograms $\boldsymbol{q}_k^{n,m} = \{\boldsymbol{q}_t^{n,m} : t_{k-1} \leq t \leq t_k\}$. Each $\boldsymbol{q}$ is a vector of H-bin histogram, and

$$\boldsymbol{q} = \{q_h\}_{h=1\ldots H}, \quad \sum_{h=1}^{H} q_h = 1, \tag{15}$$

and the mean of $\boldsymbol{q}_k^{n,m}$ is denoted by $\boldsymbol{\mu}_k$.

Assume that there are L collections of measurements acquired from other sensors before t_{k-1}. They are listed as candidates to be compared with the measurements $\boldsymbol{Y}_k^{n,m}$. For ease of exposition, $\boldsymbol{Y}_k^{n,m}$ is simply denoted as $\boldsymbol{Y}_k$, and other L collections of measurements are denoted as $\{\boldsymbol{Y}_l\}_{l=1\ldots L}$. The corresponding object of $\boldsymbol{Y}_k$, as previously mentioned, enters R_n from state Z_i. Measurements $\boldsymbol{Y}_l$ are selected as matching candidates, since their corresponding objects exits from those states $\{Z_l\}$ that are possible previous states of state Z_i, that is, there are paths in the activity graph that connect state $\{Z_l\}$ to Z_i.

Color histograms $\boldsymbol{q}_k^{n,m}$ are compared to those of other candidate objects in order to determine its identity, that is, how likely object m is the candidate object based on the measurements $\boldsymbol{Y}_k$ and $\{\boldsymbol{Y}_l\}_{l=1\ldots L}$. Assume that the collection of measurements $\{\boldsymbol{Y}_l\}$ corresponds to object l. We compute $s_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}}$, the similarity of each $\boldsymbol{q}_t^{n,m}$ to $\boldsymbol{\mu}_l$, the mean value of $\boldsymbol{q}_l$, based on a histogram intersection algorithm [Swain and Ballard 1991]. Intuitively, the similarity between two color histograms of the same object should be much larger than those of different objects. Let W be the set of similarities $\{s_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}} : t_{k-1} \leq t \leq t_k\}$; we compute $d_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}}$, the distance between $\boldsymbol{q}_t^{n,m}$ and $\boldsymbol{\mu}_l$, as

$$d_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}} = 1 - \frac{s_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}} - \min(W)}{\max(W) - \min(W)}. \tag{16}$$

Similar to the softmax function, the probability of the new observation labeled as l given its $t_k - t_{k-1} + 1$ samples of color histograms is

$$p(m = l|\boldsymbol{q}_k^{n,m}) = \frac{\prod_{t=t_{k-1}}^{t_k} \exp(-d_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}})}{\sum_{l'=1}^{L} \prod_{t=t_{k-1}}^{t_k} \exp(-d_{\boldsymbol{\mu}_{l'},\boldsymbol{q}_t^{n,m}})}. \tag{17}$$

This measure is the association probability, and we use it as an approximation to $\mathbb{E}(\gamma_{li})$, that is, the probability of an object leaving Z_l entering state Z_i, as described in the next section.

3.5. Learning Under Identity Uncertainty

Although we cannot directly solve for $\hat{p}_{ij}$ and $\hat{\lambda}_{ij}$ in Equations (8) and (14), as we do not know the identities of objects, we can use the association probabilities just computed (as in Equation (17)) to resolve this issue. Instead of maximizing the log likelihood to estimate the parameters, we maximize the expected complete log likelihood. The

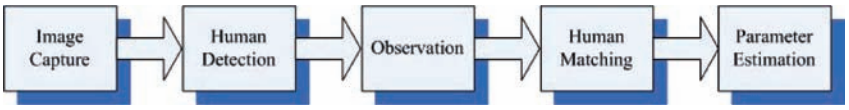


Fig. 3. A system overview.

expected complete log likelihood for the transition probabilities is

$$\begin{aligned}\mathbb{E}[\mathcal{L}(p)] &= \mathbb{E}\left[\log\left(\prod_{k=1}^{N_e}\prod_{j=1}^{S} p_{ij}^{\gamma_{kj}}\right)\right]\\ &= \mathbb{E}\left[\sum_{k=1}^{N_e}\sum_{j=1}^{S}\gamma_{kj}\log(p_{ij})\right]\\ &= \sum_{k=1}^{N_e}\sum_{j=1}^{S}\mathbb{E}(\gamma_{kj})\log(p_{ij}),\end{aligned}$$

which can be solved using the estimates described in the previous section. Therefore, our estimates are

$$\hat{p}_{ij} = \frac{\sum_{k=1}^{N_e}\mathbb{E}(\gamma_{kj})}{\sum_{k=1}^{N_e}\sum_{j=1}^{S}\mathbb{E}(\gamma_{kj})}. \tag{18}$$

Similarly, the traveling times can be estimated as

$$\frac{1}{\hat{\lambda}_{ij}} = \frac{\sum_{k=1}^{N_e}\mathbb{E}(\gamma_{kj})t_{ij}}{\sum_{k=1}^{N_e}\mathbb{E}(\gamma_{kj})}. \tag{19}$$

Note that our approach resembles the expectation-maximization (EM) algorithm in which the computation of the association probabilities $\mathbb{E}(\gamma_{kj})$ is the E-step and the parameter estimation is the M-step. However, no iteration is required in our formulation, since the results from the M-step do not affect the computation of association probabilities. Nevertheless, it is possible to incorporate traveling times into association probabilities, and then the EM algorithm can be used to estimate the parameters.

4. APPLICATION: LEARNING TRAFFIC PATTERN IN A SMART BUILDING

4.1. Overview

To validate our proposed framework, we carry out experiments in a smart building equipped with a network of CITRIC camera motes for modeling and predicting traffic patterns of dwellers. Figure 3 shows an overview of the whole system. In our system, the images are first captured at the resolution of 320×240 pixels, which provides sufficient visual information for human detection. The captured images are then processed through an HOG-based human detection module to extract useful features (observations) for human matching. The observations are the normalized RGB histograms of the upper torso of detected image regions. In the training phase, we estimate the model parameters, $\{p_{ij}\}$ and $\{\lambda_{ij}\}$ as described in Section 3, using the images of dwellers detected from the motes. Once a subject is detected, its next state and expected arrival time can be predicted using the learned traffic model.

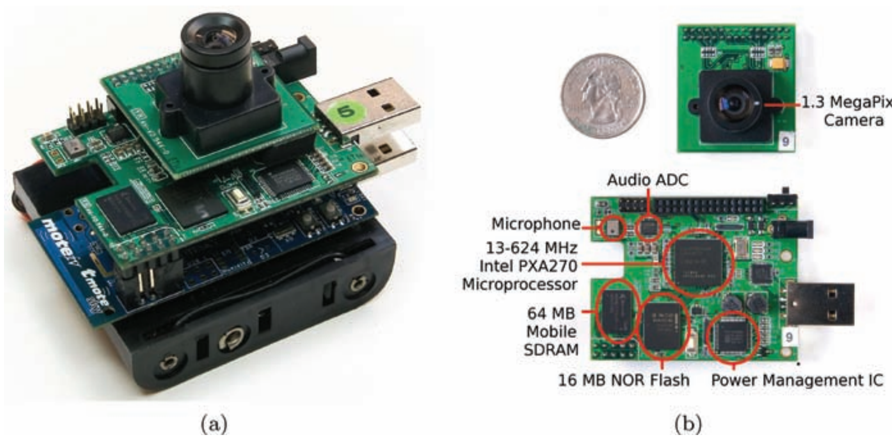


Fig. 4. CITRIC camera mote. (a) An assembled camera daughter board with Tmote Sky board. (b) A camera daughter board with major functional units outlined.

4.2. Platform: CITRIC Mote

Our experiments are carried out using a network of lightweight CITRIC motes [Chen et al. 2008]. The CITRIC mote is a wireless camera system consisting of a camera daughter board and a Tmote Sky board. The camera daughter board is equipped with a CCD camera, a frequency-scalable (up to 624MHz) CPU, 16MB FLASH, and 64MB RAM (see Figure 4). The CITRIC mote uses the OmniVision OV9655 CMOS image sensor [Omnivision Technologies Incorporated 2006], which offers the full functionality of a camera and an image processor on a single chip supporting various capture modes (e.g., SXGA, VGA, and CIF). It is able to capture images up to 30 frames per second in VGA and CIF modes, and 15 frames per second in SXGA mode.

The computing module of CITRIC consists of an Intel fixed-point PXA270 processor [Intel Corporation 2004] with a maximum speed of 624MHz, 256KB of internal SRAM, and a wireless MMX coprocessor to accelerate multimedia operations. The PXA270 processor is equipped with the Intel Quick Capture Interface that eliminates the need for external preprocessors to connect the processor to a camera sensor. The PXA270 processor is connected to 64MB of 1.8V Qimonda Mobile SDRAM and 16MB of 1.8V Intel NOR FLASH memory. The SDRAM is used for storing image frames, and the FLASH memory is for storing code.

The camera daughter board uses the Silicon Laboratories CP2102 USB-to-UART bridge controller to connect the UART port of the PXA270 processor with a USB port on a personal computer for programming and data retrieval. The CITRIC mote also provides wireless communications over the IEEE 802.15.4 protocol, which makes it convenient to collaborate with other motes.

As it provides more computing power and compact integration of physical components with relatively little power consumption, the CITRIC mote enables a wider variety of distributed pattern recognition applications and in-network processing of images to reduce communication bandwidth requirements.

4.3. Algorithm Implementation

In this section, we will explain the implementation detail of each module, as shown in Figure 3.

4.3.1. Image Capturing and Human Detection. The image frames are captured and saved on the CITRIC motes at the rate of four frames per second. All the raw images are transmitted to a central server for offline training. As only those frames with human are informative in our experiments, we apply a detector based on histogram of oriented gradients (HOG) [Dalal and Triggs 2005] to detect humans in these images, where the outputs are their coordinates in the scenes.

When one subject walks across the FOV of a camera, multiple frames of the subject will be captured by the camera. Consequently, even if a subject is not detected in some frames (i.e., false negatives), the effects of these errors on final results are negligible.

As the camera positions are fixed, we can exploit prior spatial and temporal knowledge of human subjects to eliminate most of the false positives. For example, we know a priori that no person would appear in the air when walking, and thus any detected results violating this rule are false positives and can be removed. Furthermore, we can also remove some false positives based on temporal consistency. As we have consecutive frames, if at some frame the detection result (i.e., the coordinates of the bounding box) deviates significantly from the results of other frames, we can remove this frame as either it is a false positive or there is another subject in that position. In both cases, the detection results from such frames can be removed so that the traffic patterns can be better modeled. We will give more results in Section 4.4.

It is worth mentioning that human detection results provide more useful information than methods with simple background subtraction with blob models. For example, multiple humans can be detected and differentiated in a scene, thereby facilitating flow analysis of groups (see Section 4.5.2 for discussion).

4.3.2. Measurements. In this work, the camera motes are assumed time synchronized, and a unique time stamp is assigned to each image frame from all five cameras. The time stamps and image coordinates from human detection provide strong cues for inferring which frames belong to the same subject based on all sequences acquired by these cameras. For each detected subject, the entering time t_- and the leaving time t_+ of a scene are recorded.

Image coordinates of the detected subject at the entering/leaving time and the direction of movement help in determining the entering state and leaving state of one subject, as the placements of cameras are approximately known (i.e., their relative topology). That is, the expected size and position of a detected human with respect to a camera can be exploited for inference. For example, at camera C_1 in Figure 5(a), if the subject is observed entering from the left-hand side of the scene, then the entering state must be Z_1. If the subject is observed leaving from the far right end (with smaller bounding box), the leaving state is Z_3. Otherwise, if the bounding box is large and detected on the right side of the frame observed from C_1 when the subject leaves the scene, its leaving state is Z_2.

Besides the time stamps and state information, another measurement is the image observation defined in Equation (3). After fitting an ellipse in the bounding box of a detected subject to remove background pixels, we compute the normalized RGB histogram for the foreground part. The normalized RGB color histogram is used for image representation, as it is invariant to change in scale and viewpoint, which in turn facilitates the matching process.

4.3.3. Human Matching. We exploit both spatial and temporal prior information for matching between clusters. In this work, a cluster is defined as a collection of frames captured by one camera within a time period that belongs to one subject. For example, in Figure 5(a), if a subject enters R_1 at Z_1 and exits at Z_3 after some time, then all the frames captured by C_1 during this time belong to one cluster. If the same subject enters Z_3 at some other time stamps, those captured frames are put in another cluster. Clusters are processed and matched in chronological order. First, with prior spatial knowledge of camera placements and structure constraints, we know all the possible

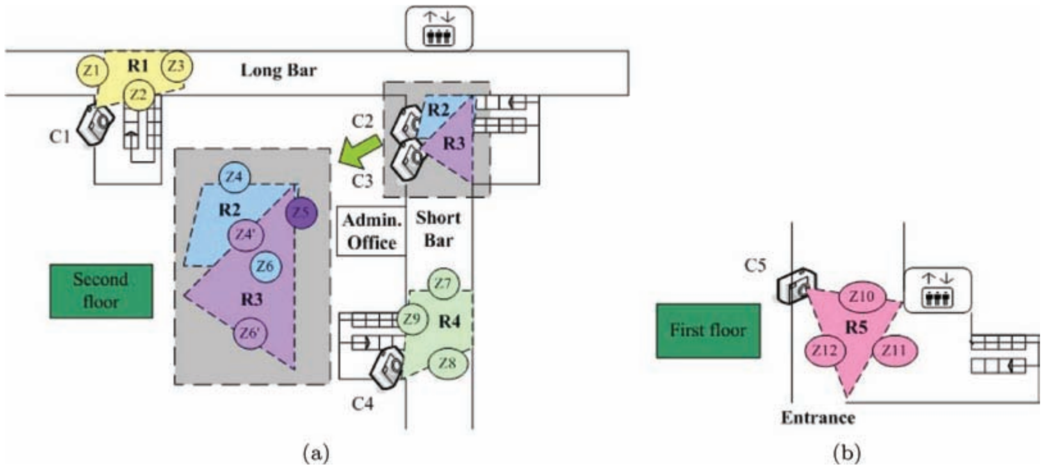


Fig. 5. Experimental setup. (a) Four camera motes are placed on the second floor of a building. The FOV (colored region) of each camera has a different shape, and Z_1 to Z_9 are possible entrance/exit points (states) of each region (R_2 and R_3 are shown in a larger region on the lower left). Note that the sensing region of cameras C_2 and C_3 are overlapped. (b) One camera is placed on the first floor. A person passing state Z_{11} can either take the elevator or the stairway to the second floor, thereby reaching state Z_3 (out of elevator and turn right immediately), Z_4 (out of elevator and walk straight), or Z_5 (take stairway and reach Z_5).

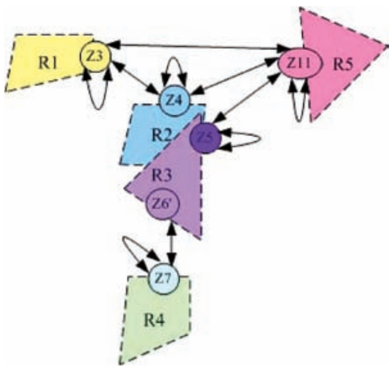


Fig. 6. Activity graph representation in our specific experimental setting. Selected states are shown. Only those trajectories starting from or ending into states Z_3, Z_4, Z_5, Z_6, Z_7, and Z_{11} contribute to the traffic pattern model.

state transitions. For example, as seen in Figure 6, the possible next states for Z_4 are $\{Z_0, Z_3, Z_4, Z_{11}\}$. We define another state Z_0 to account for situations when $\forall j, \gamma_{kj} = 0$ (defined in Equation (4)), that is, the subject k does not enter any state Z_j after exiting Z_i. Thus, Z_0 is a "sink" state which accounts for the areas not observed by all cameras (i.e., there are some blind spots not covered by cameras). When a new subject is first detected in the scene, it is considered to start from state Z_0. Likewise, a subject arrives at state Z_0 when it is last detected by any camera.

Assume that at time t_{-}, there is a detection by camera n. As previously mentioned, we can infer the entering state of a subject, say, Z_i, from the coordinates of the bounding box. Let the list of possible previous states of Z_i be $E_i : \{Z_{i1}, \ldots, Z_{im}\}$, where m is the total number of possible state transitions into Z_i. It follows that only the image clusters within a time window associated with those states in E_i are considered for matching. The threshold for the time window is determined based on the prior knowledge of camera placements (e.g., larger threshold values for two states with long distance or connected via an elevator) and the typical speed of moving subjects. Let A_i denote the set of all possible clusters satisfying the spatial and temporal constraints. If A_i is empty, it means there are no other suitable clusters to compare with, and the subject is regarded as a new person in the scene. If A_i is not empty, we first compute the distances between the image cluster at Z_i and each cluster in A_i. The corresponding distances and matching probabilities are computed according to Equations (16) and (17). If all the matching probabilities are relatively small, the subject is regarded as a new person appearing from some blind spots, that is, entering the scene from Z_0. The corresponding subject is considered as disappearing in the scene, that is, arriving at state Z_0, if there is no matched cluster from A_i.

4.4. Experimental Setup and Results

4.4.1. Setup. Figure 5 shows the building layout and the placements of CITRIC motes. Five CITRIC motes are placed on two floors of a building at the intersections of hallways as well as stairways, with four on the second floor, and the other one on the first floor. The FOVs of the five cameras are denoted by R_1 to R_5, and their corresponding states are denoted by Z_1 to Z_{12}, as shown in Figure 5 (where R_2 and R_3 are shown with larger images). Four cameras are placed on the second floor near the stairways, and the other one is placed on the first floor near the entrance. Cameras are deployed on the side of the hallways to capture full images of humans. The states are determined based on the approximate camera position and their FOVs. In general, one possible entry/exit point corresponds to one state. As constrained by the physical structure of the building, the sensing regions have different shapes, and some states are covered by more than one region (e.g., Z_5 is covered by R_2 and R_3). As the states represent the entry and exit points of a region, it is easy to see that states Z_4 and Z_4' are actually connected seamlessly, that is, subjects walking through state Z_4 will definitely arrive at state Z_4'.[2] Therefore, we consider them as one state (likewise for Z_6 and Z_6') in the following discussions.

The activity graph for this experimental setting is shown in Figure 6. Prior knowledge of the building layout is used to determine the connectivity between states. Most states are connected by paths through corridors. Specifically, Z_{11} is connected to Z_3 and Z_4 via elevator. That is, a person detected by C_5 in R_5 is likely to appear in R_1 and detected by C_1 if the person takes the elevator and walks directly toward R_1 (note that the sensing region of C_2 does not cover the corridor region right in front of the elevator), or R_2 (and detected by C_2) if the person walks toward R_2 after taking the elevator or stairway. Likewise, Z_5, is connected to Z_{11} as a subject may take the stairway from the first floor and walk toward R_2. These paths in the activity graph match real-world traffic patterns of dwellers in this building.

Not all possible trajectories will contribute to the traffic pattern model, as we only consider the traffic pattern between $\{R_n\}$'s. For example, it does not matter where the subjects come from before entering state Z_1 or Z_2. We define another state Z_0 to account for all the regions not covered by $\{R_n\}$'s, as described in Section 4.3.3.

4.4.2. Experiment I: Human Detection Using CITRIC Motes. We describe the implementation of the histogram of oriented gradients (HOG) for human detection using a CITRIC mote. In the HOG feature representation, an image patch is equally divided into nine rectangular grids in which each one is represented by the image gradients (see Figure 7). It has been demonstrated that the HOG-based detector performs well in detecting human subjects from cluttered images [Dalal and Triggs 2005].

[2]Considering that it is rare that one subject will enter state Z_4 and return before reaching state Z_4'.

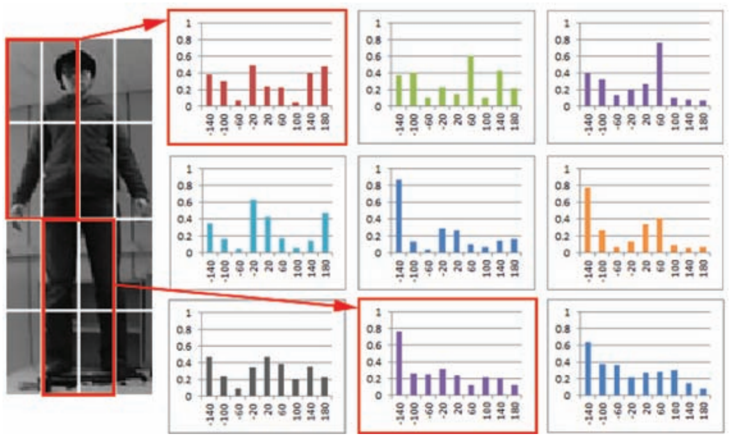


Fig. 7. A HOG descriptor. An image patch is divided into 4 × 4 cells. A section is constructed from 2 × 2 cells. For each section, a histogram of gradients is calculated. There are nine bins in each histogram which divides 360° evenly (each bin represent 40°). Hence, each HOG descriptor contains 81 values (9 histograms × 9 bins).

The HOG-based human detection algorithm requires two major operations: computation of a HOG descriptor and classification using, for instance, a support vector machine (SVM). Both operations require heavy floating point operations. However, the CITRIC mote is equipped with a PXA270 processor which does not have a physical floating-point unit. Hence, any floating-point operations must be carried out using software emulation, slowing down the overall processing time. As shown next, it requires about 355 ms for the two operations. However, we are in the process of developing the next version of a CITRIC mote with a physical floating point unit, and it is expected that the new platform can reduce the computation time for the two operations below 100 ms and achieve the minimum frame rate of four frames per second for reliable detection of humans.

Since it requires at least four frames per second for reliable detection of humans for traffic modeling, we could not run the full experiment using CITRIC motes. Instead, we demonstrate the potential of our proposed approach by showing the performance of the HOG-based human detection on a CITRIC mote (with software floating-point emulation) and experimental results on traffic modeling using images captured by CITRIC motes.

We first describe the implementation of the HOG on a CITRIC mote using software floating-point emulation. In order to save computation time, we first detect the foreground region from which the HOG descriptor is computed. Then, SVM is trained[3] to classify whether the detected foreground is a human or not. The overall flowchart of these operations is shown in Figure 8.

We attach a CITRIC mote on the wall of a hallway to gather a dataset. A foreground object is detected using an image difference method. If the number of pixels in the detected foreground object is larger than a certain threshold, the region is considered as foreground area. In order to suppress false detections, the system allows a user to select a region of interest, as shown in Figure 9(b). With a region of interest, we can easily ignore regions with high non-human activity, such as doors that are opening and closing, as shown in Figure 9(a). We collect 631 images with 348 positive samples and 283 negative samples. See Figures 10 and 11 for some examples from which our SVM classifier is trained.

[3]SVM light. `http://svmlight.joachims.org/`.

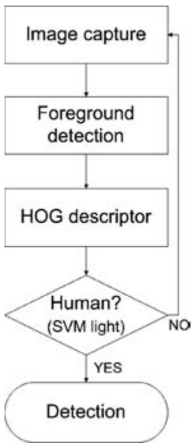


Fig. 8. A flowchart of a HOG-based human detector on a CITRIC mote.

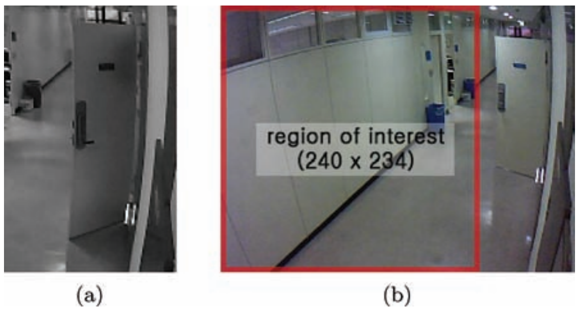


Fig. 9. When a whole image is searched for foreground objects, it requires scanning more windows than necessary. For example, the door area can be excluded for computational efficiency (a). This problem can be addressed by labeling a region of interest for detecting foreground objects, as shown in (b). Restricting our attention to a region of interest improves performance in human detection and, at the same time, saves unnecessary computation.

Fig. 10. Examples of positive samples. Faces are masked for privacy.

We additionally collect 199 images for tests (137 positive cases and 62 negative cases) to evaluate our HOG-based human detector using a CITRIC mote. The resulting precision and recall plot is shown in Figure 12, where we vary the threshold value of SVM. The figure shows that we can reliably detect a human with precision of over 80%. The actual operating point can be selected based on the type of application at hand. The computation time of each operation of the HOG-based human detector is shown in Table I. Since the sizes of detected foreground objects are different, we compute the

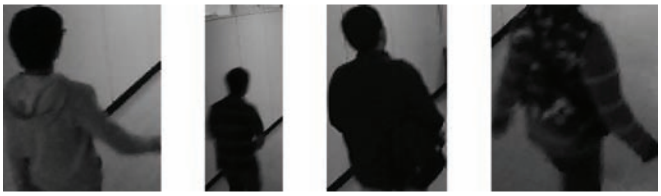


Fig. 11. Examples of negative samples. Parts of a person and people with unusual poses are considered as negative samples for detecting a full-body person.

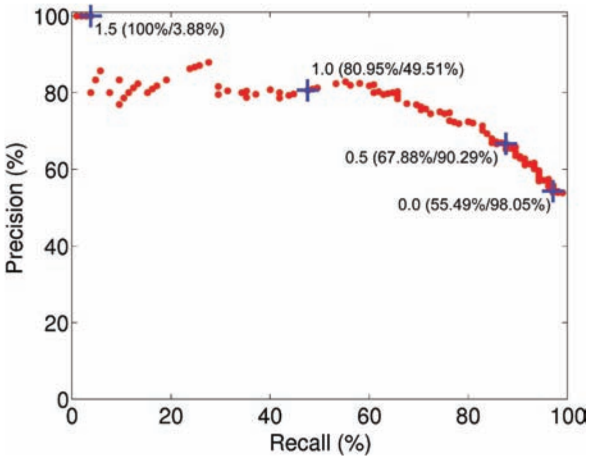


Fig. 12. A precision-recall plot of the HOG-based human detector on a CITRIC mote. Each point in the plot is computed by running SVM*light* over the whole test set at a different threshold value. Four notable threshold values at 1.5, 1.0, 0.5, and 0.0 are shown with plus marks along with precision/recall percentages in parentheses.

Table I. Computation Times of Operations of the HOG-Based Human Detector on a CITRIC Mote (in *ms*)

Operation type	Average computation time (*ms*)	Standard deviation (*ms*)
Image capture	48.98	1.68
Foreground detection	3.48	0.55
Cropping the foreground region	2.35	1.28
Memory allocation for HOG	99.55	37.19
HOG descriptors	20.62	13.98
SVM	234.84	4.80

average and standard deviation of the computation time of each operation of the HOG-based human detector. The operations for HOG and SVM uses over 85% of the total computation time. This is due to the fact that floating-point operations are enumerated in software. We expect that we can dramatically reduce the computation time for HOG and SVM operations in our next generation of CITRIC motes.

4.4.3. Experiment II: Traffic Modeling and Prediction. In this section, we will present experimental results, including the model parameter estimates to validate our proposed framework. At each camera, two image sequences are collected with more than 40 people walking through this building. Over 100,000 images frames are captured and saved by these motes over several days. From these frames, 14,000 images containing humans are detected. Overall, the HOG-based detector performs well in our experiments

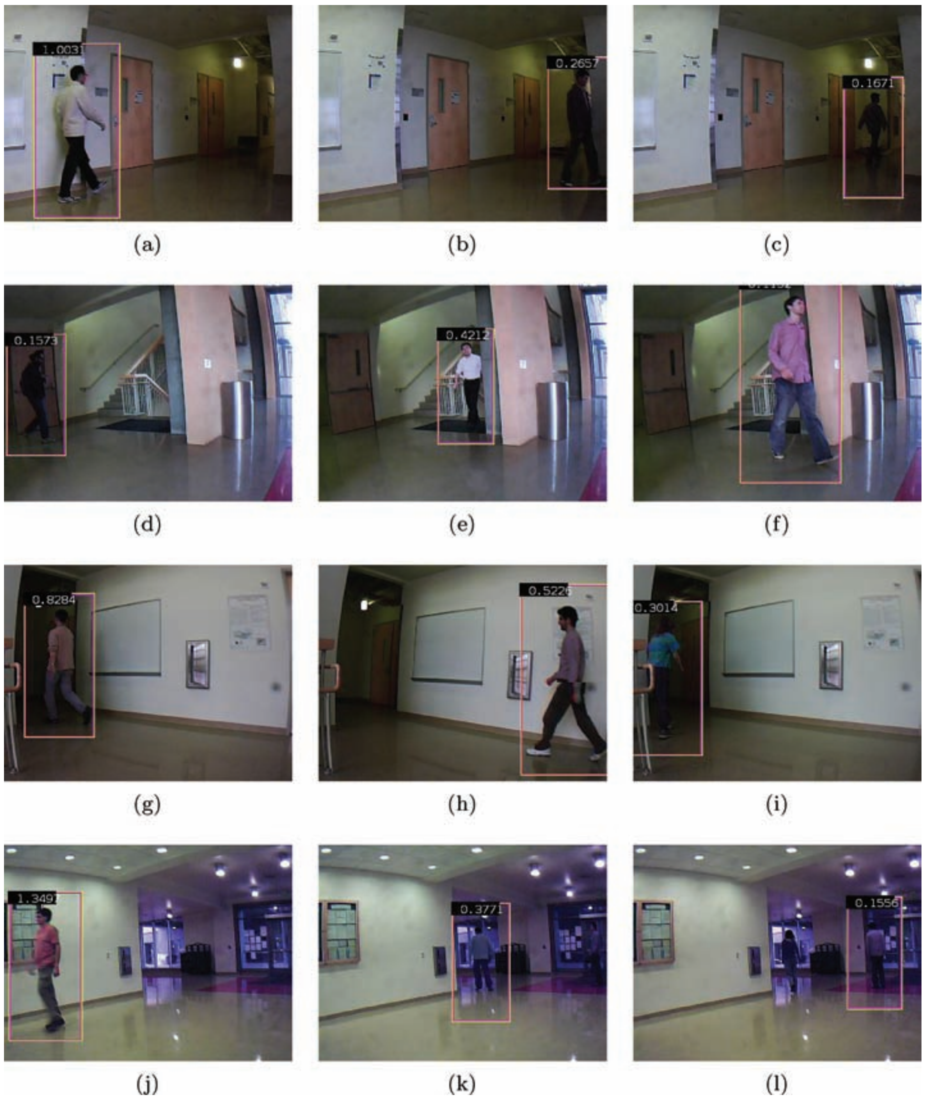


Fig. 13. Human detection results. Images (a)–(l) are shots at states Z_1 to Z_{12}, respectively. The detected subjects appear in different viewpoints and sizes and under different lighting conditions, which makes the object association problem rather challenging. The image coordinates of the bounding box help identify the entering and leaving states of the subjects.

with few false negatives and false positives. Figure 13 shows some human detection results. The detected subjects appear in images acquired at different viewpoints and distances under different lighting conditions. The values at the top-left corners of the bounding boxes represent the response (i.e., confidence) of the HOG-based detector. Of all the 104,948 images collected, there are 13,577 detections with only 792 false positives and 1,749 false negatives. The true positive rate is 88.59%, while the false positive rate is 0.88%. We partition the image frames into two sets, with 9,003 frames for training and 4,574 frames for testing. Some of the videos and results are available at `https://eng.ucmerced.edu/people/zshuai/tosn.html`.

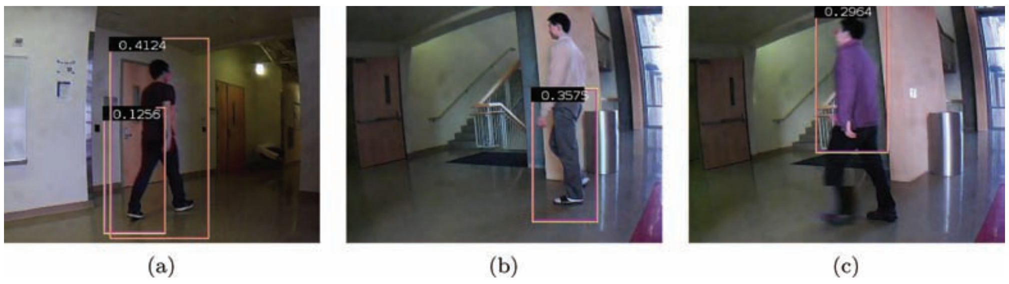


Fig. 14. Some false positives from our HOG-based detector. Most of the false positives can be removed using prior spatial knowledge. (a) One bounding box is embedded in the other. (b) The y_{min} coordinate of the bounding box is too large (i.e., the detected person is too small). (c) The y_{max} coordinate of the bounding box is too small (i.e., the detected person does not walk on the floor).

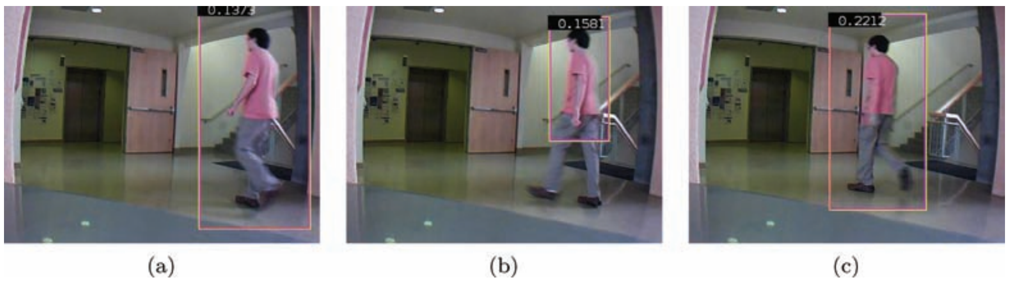


Fig. 15. Human detection results from three consecutive frames. (a) and (c) are true positives, but (b) is a false positive, which can be easily identified and removed by maintaining temporal consistency of their bounding box coordinates (i.e., a person is very unlikely to impulsively jump to the ceiling while walking).

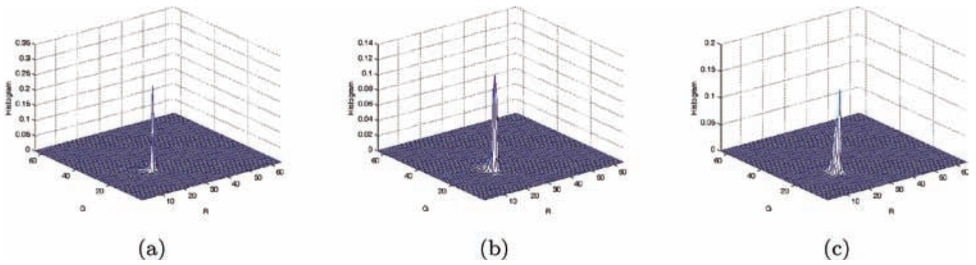


Fig. 16. Representative normalized RGB histograms. The x- and y-axis stand for the R, G channel, respectively. The components are mainly in (10:30, 10:30) range, and we use the 20 × 20 bins to represent human.

Some false positive examples are shown in Figures 14[4] and 15. Figure 14 illustrates how to remove false positives with prior spatial knowledge, and Figure 15 shows an example of applying temporal constraint to remove false positives.

We use normalized RGB histogram as the observation (measurements). The original normalized RGB histogram has 64 × 64 bins (64 bins for R and G channels, respectively). In the experiments (as shown in Figure 16), we found that the components are mainly in a small region (i.e., (10:30, 10:30) in both normalized R and G channels).

[4]Let the upper-left corner of the image be the origin. The coordinates of the bounding box are fully determined by four values: $\{y_{min}, y_{max}, x_{min}, x_{max}\}$, which can be compared with the image coordinates of floors.

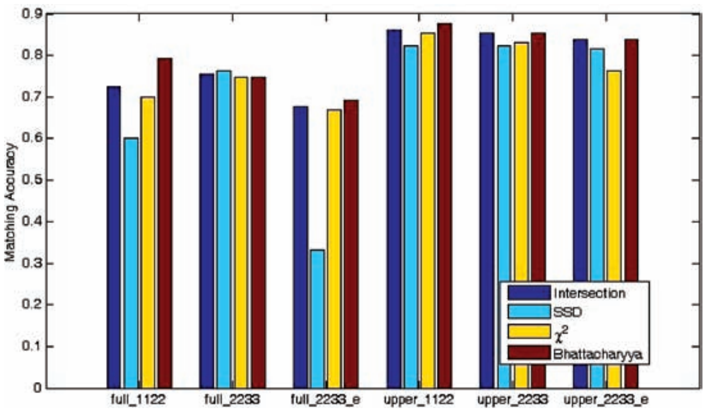


Fig. 17. Comparisons using different matching metric, ellipse size, and upper-body/full-body color histograms. The four-digit labels in the x-axis refer to the size of the ellipse fit in the bounding box, for example, 1122 means the distances from the four vertices of the ellipse to $y_{min}, y_{max}, x_{min}, x_{max}$, respectively, is $0.1\times$, $0.1\times$ the height of the bounding box h, and $0.2\times$, $0.2\times$ the width of the bounding box w. From left to right, the x-axis labels indicate (1) full body, ellipse size $0.8h \times 0.6w$, no ellipse. (2) full body, ellipse size $0.6h \times 0.4w$, no ellipse, (3) full body, ellipse size $0.6h \times 0.4w$, fit ellipse, (4) upper body, ellipse size $0.8h \times 0.6w$, no ellipse, (5) upper body, ellipse size $0.6h \times 0.4w$, no ellipse, (6) upper body, ellipse size $0.6h \times 0.4w$, fit ellipse.

To reduce computational cost, we therefore use 20×20 bins for the normalized RGB histogram to represent image observations of humans.

We have tested other representations, including conventional RGB histogram, kernel RGB histogram, and gray-scale histograms. We have also conducted experiments to evaluate the matching performance using the histogram from the upper torso (upper-part of fitted ellipse) or from the entire body (the whole ellipse). Figure 17 shows the matching accuracy conducted on the collected dataset. Our results show that the matching accuracy using the color histogram of an upper body is constantly better than that with a full-body histogram. It may be explained by the fact that the color of the lower torso (e.g., color of pants) has less variation compared to the upper part. In addition, the lower part of the ellipse inevitably includes some background pixels, which further reduces the identification capability. We have experimented with various representation schemes and find that the one using the normalized RGB histogram of the upper human torso with 20×20 bins performs best with the matching metric based on histogram intersection [Swain and Ballard 1991].

We have also carried out a series of experiments to find an appropriate matching metric including histogram intersection, Bhattacharyya distance [Forsyth and Ponce 2002], chi-square distance, sum of squared differences (SSD), and earth mover distance (EMD) [Rubner et al. 2000]. Some of the results are shown in Figure 17. While the results using Bhattacharyya distance are slightly better than those with histogram intersection in these experiments, the method with Bhattacharyya distance involves multiplication and square root (floating) operations, and the one with histogram intersection requires only comparison and addition (integer) operations. Considering the ease of implementation and its runtime performance, the method with histogram intersection is a better option due to its efficiency and effectiveness.

4.4.4. Training Phase. Figure 18 shows some example sequences used in the training phase in which each trajectory shown in a different color describes one possible path. For example, the leftmost green trajectory represents one subject who is first captured by C_1 moving from state Z_1 to Z_3. After walking along the long corridor, the subject ends at state Z_4 and enters R_2 and R_3. The subject finally leaves Z_6 and reaches state

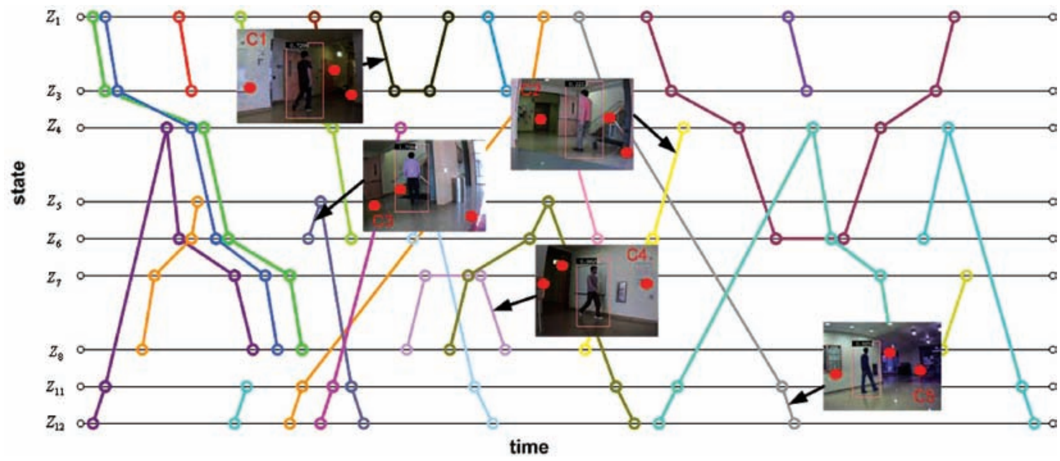


Fig. 18. Sample sequences used in our experiments. The x-axis and y-axis represent time and state (entry/exit node), respectively. The trajectories of different subjects are shown in solid lines of different colors, and the solid red dots denote the states. Sample images acquired at five cameras are also shown in the figure.

Z_7. These trajectories can be long or short, and the purple line on the top right corner is an example of short trajectory. After leaving Z_3, the corresponding subject ends in the region not covered by $\{R_n\}$'s. As there are many offices and labs along the long corridor, the subject may have entered one of these. The five small figures are selected shots by five cameras, and the red dots denote the states. These paths indicate that subjects move freely in various patterns that and the subjects are observed by up to four cameras. Note that not all the states are shown in the figure, as some of them do not contribute to the traffic model, for example, Z_2 and Z_9 (where subjects enter regions not monitored by the cameras). It is worth noting that these trajectories can be identified and matched through the images acquired at different cameras by our algorithm.

Figure 19 shows some detection results from images captured by different cameras (where the detected results are normalized to a canonical size). Note that images of subjects in various pose can be detected by our method. Note also that appearances of the same subject may change dramatically, as viewed by different cameras, due to variation of lighting and the responses of CCD sensors.

As the goal here is to model and predict the traffic patterns of all dwellers in a building, we need to estimate the transition probabilities of all states from all recorded sequences. The state transition probability and traveling time can be estimated, as described in Section 3.

Experimental results using the training set are shown in the second and third columns of Table II. The second column lists the estimated state transition probabilities ($p_{i\to j}$) with expected traveling times ($t_{i\to j}$, i.e., $\frac{1}{\lambda_{i\to j}}$, same as in Equation (19)) in parenthesis of the training phase; the third column presents the corresponding ground truth. The ground truth values are obtained by visual inspection of all the frames to determine the matched subjects and by counting the frequency of how the subjects move between states. Overall, these estimated probabilities and traveling times of our model match the ground truth values well. The average error for all state transition estimates is 0.0602, and the standard deviation is 0.0791.

There are a few cases in which our model does not estimate the state transition probabilities well. For example, from the ground truth data, we know there is no

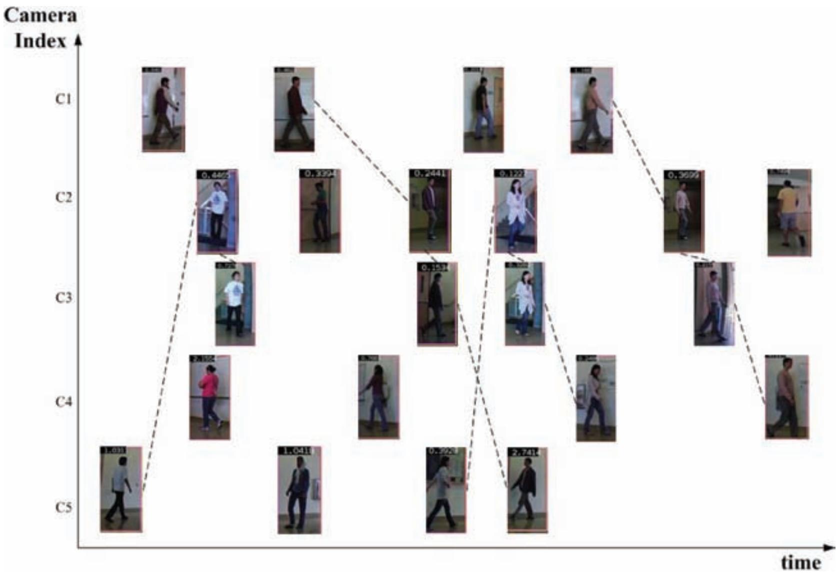


Fig. 19. Some detection results and example trajectories. The x-axis and y-axis represent the time and camera index.

Table II. Estimated Model Parameters and Ground Truth

State Transition	Estimated Parameters	Ground Truth from training phase	Ground Truth from validation phase
$p_{3\to0}$ ($t_{3\to0}$)	0.3082 (–)	0.4 (–)	0.3333(–)
$p_{3\to3}$ ($t_{3\to3}$)	0.1177 (19.05)	0.1 (22)	0.1111 (28)
$p_{3\to4}$ ($t_{3\to4}$)	0.4384 (36.18)	0.4 (35)	0.4444 (32.75)
$p_{3\to11}$ ($t_{3\to11}$)	0.1357 (56.59)	0.1 (75)	0.1111 (91)
$p_{4\to0}$ ($t_{4\to0}$)	0.3197 (–)	0.4 (–)	0.4444 (–)
$p_{4\to3}$ ($t_{4\to3}$)	0.4372 (18.86)	0.3 (18)	0.3333 (31.33)
$p_{4\to4}$ ($t_{4\to4}$)	0.0052 (28.83)	0 (–)	0 (–)
$p_{4\to11}$ ($t_{4\to11}$)	0.2379 (52.37)	0.3 (51)	0.2222 (48)
$p_{5\to0}$ ($t_{5\to0}$)	0.2701 (–)	0.3 (–)	0.3 (–)
$p_{5\to5}$ ($t_{5\to5}$)	0.1298 (12.21)	0.1 (12)	0.2 (22.5)
$p_{5\to11}$ ($t_{5\to11}$)	0.6001 (26.77)	0.6 (29)	0.5 (26.8)
$p_{6\to0}$ ($t_{6\to0}$)	0.2863 (–)	0.5 (–)	0.5714 (–)
$p_{6\to6}$ ($t_{6\to6}$)	0.1931 (19.67)	0.1 (22)	0.1429 (20)
$p_{6\to7}$ ($t_{6\to7}$)	0.5206 (18.70)	0.4 (15.5)	0.2857 (21.5)
$p_{7\to0}$ ($t_{7\to0}$)	0.1850 (–)	0.1818 (–)	0.1429 (–)
$p_{7\to6}$ ($t_{7\to6}$)	0.5041 (11.83)	0.5455 (11.25)	0.5714 (28)
$p_{7\to7}$ ($t_{7\to7}$)	0.3109 (18.99)	0.2727 (20)	0.2857 (22.5)
$p_{11\to0}$ ($t_{11\to0}$)	0.2741 (–)	0.25 (–)	0.25 (–)
$p_{11\to3}$ ($t_{11\to3}$)	0.1451 (52.75)	0.1667 (57.5)	0.125 (83)
$p_{11\to4}$ ($t_{11\to4}$)	0.1711 (30.25)	0.25 (24)	0.25 (39.5)
$p_{11\to5}$ ($t_{11\to5}$)	0.2905 (24.68)	0.3333 (22.25)	0.375 (30.67)
$p_{11\to11}$ ($t_{11\to11}$)	0.1192 (25.29)	0 (–)	0 (–)

subject moving from Z_{11} to Z_{11}, but the estimated probability of moving from Z_{11} to Z_{11} is 0.1192 with an average traveling time of 25.29 seconds. This error results from false matching results, and this effect is expected to be negligible when a large dataset is used. Furthermore, as shown in Figure 5, two adjacent cameras, C_2 and C_3 have overlapped FOVs, and thus most subjects appearing in region R_2 and R_3 are likely to be observed by both cameras. Instead of using the images acquired from one camera, it is likely to have fewer matching errors by exploiting such information.

4.4.5. Validation Phase. We validate the estimated traffic model with test sequences where the ground truth is obtained by visually inspecting the trajectories of all the subjects. The experimental results with the test sequences are shown in the fourth column of Table II. It is clear that the estimated model is able to predict the transition probabilities and traveling time well. The estimated state transition probabilities and traveling time using the training set do not vary too much from those using the validation set. In general, the errors are slightly larger than those in the training sequences, which is what we expected. These results indicate that the learned model parameters do not overfit the training data.

4.5. Discussion

4.5.1. What Can We Infer From the Traffic Model?. The results listed in Table II match our expectation well. For example, for all the objects leaving from state Z_3, there is a high probability that they will enter state Z_0 or Z_4. As shown in Figure 5, the states Z_3 and Z_4 denote areas which are connected by a corridor full of offices and labs. Thus, people may enter this corridor from the left (state Z_3) and enter offices or labs (some regions between areas denoted by Z_3 and Z_4). In addition, the administration office is located at a room between regions represented by Z_6 and Z_7. This explains why people leaving state Z_3 are likely to arrive at state Z_4 with high probability, as Z_4 and Z_6 are two states which must be visited from Z_3 before arriving at the administration office (between states Z_6 and Z_7).

By examining the transition probability at state Z_5, more than half of the out-going subjects end up at state Z_{11}, that is, most people observed at state Z_5 are likely to go downstairs and walk toward the entrance of the building. On the other hand, less than 30% of them arrive at state Z_0 (areas not monitored by our cameras). This is because the stairway also goes up to the third floor of the building (areas which are not monitored by our camera sensor network).

Another interesting observation from our traffic model is that the sum of $p_{11\rightarrow 3}$ and $p_{11\rightarrow 4}$ (two monitored areas reachable via elevator on the second floor) is larger than $p_{11\rightarrow 5}$ (one monitor area next to the stairway on the second floor), which means more people are likely to take the elevator to the second floor than the stairway, although the former way costs much more time. In addition, about 25% of the people leaving Z_{11} arrive at Z_0 (i.e., entering areas not monitored by our cameras), which happens when people take the elevator to the third floor of the building.

From the estimated traffic model, we see that the most likely trajectories of subjects starting from state Z_3 are $Z_3 \rightarrow Z_0$ and $Z_3 \rightarrow Z_4 \rightarrow Z_0$. It means people entering the building from this area are likely to enter one of the offices or labs in between the areas monitored by Z_3 and Z_4 and the administration office (between areas monitored by states Z_6 and Z_7) on the second floor. In addition, people leaving state Z_7 tend to walk toward region R_3 and go downstairs (i.e., the state transition is $Z_7 \rightarrow Z_6 \rightarrow Z_5 \rightarrow Z_{11}$).

4.5.2. Multiple Detection Case. As described earlier, multiple subjects can be identified with the HOG-based human detection algorithm. This is in contrast to prior works where tracking algorithms are used, and thus one subject is often assumed in the scene. During the training phase, there are some frames in which more than one subject are detected or the bounding box coordinates in two consecutive frames are far from each other. Figure 20 show some examples with multiple detected humans. The background image is one selected frame of the sequence. The colored rectangles represent the bounding boxes of the detection results. In Figure 20(a), one subject remains on the right side of the view and another subject walks from the left corner to right. In this sequence, the detected bounding boxes do not overlap although their

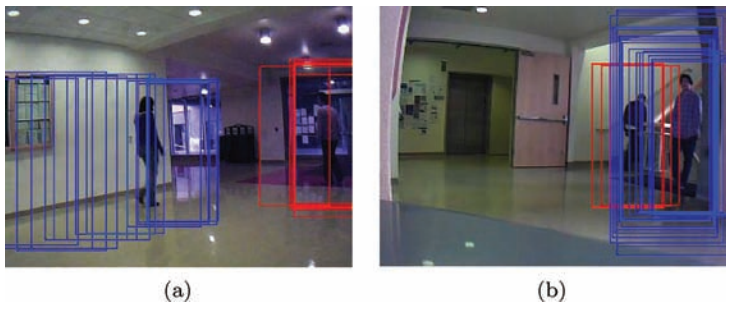


Fig. 20. Examples of multiple detected subjects in image sequences. The background image is one selected frame of the sequence. The colored rectangles represent the bounding boxes of the HOG detection results in a sequence of images. (a) The bounding boxes of two subjects do not overlap. (b) The bounding boxes of two subjects overlap with each other.

detections have similar time stamps. It is obviously incorrect if we cluster all the detected images in this sequence to one subject simply based on the observations that they are acquired at the same time interval. In fact, in the case of Figure 20(a), we can still easily group the detections into two clusters, as the bounding boxes do not overlap.

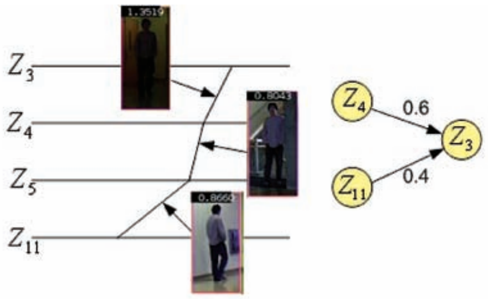
In the sequence shown in Figure 20(b), one subject first walks upstairs from the first floor to the third floor (the camera is placed on second floor), then another subject appears from the stairway and walks toward the right of the view. Although the detected bounding boxes have overlaps and are time-interleaved, we can make use of the direction cue to resolve ambiguities. For example, if the bounding boxes of the sequence first move leftwards and then suddenly to the right it is likely that these detected frames do not belong to the same cluster. We can perform local feature matching to verify whether frames belong to the same cluster or not. Assuming that the turning point happens at time t, we compute the distance between two sets of frames before and after t. If the distance is below a threshold, then the frames of these two sets belong to the same cluster, that is, the same subject.

In our experiments, we assume that only a few subjects may appear in the scene at any time, thereby facilitating the matching process per frame. Our future work will consider cases where a crowd of people moving together with more advanced vision algorithms (to detect humans under occlusion) and additional prior knowledge.

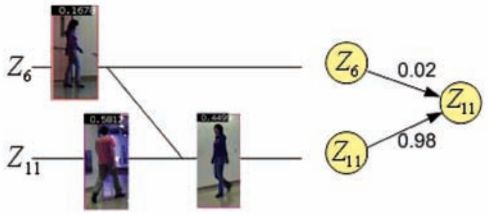
4.6. Failure Cases

In this section, we discuss a few matching failures and their effects on the estimated traffic model. Figure 21(a) shows an example where one subject walks upstairs from state Z_{11} to Z_5 and then to Z_3 via Z_4. At state Z_3, the sequences of this subject at both Z_{11} and Z_4 are selected as matching candidates (both sequences satisfy the spatial and time constraints). Obviously, two candidates are actually images of the same subject, so the similarity measure of them to the new observation at Z_3 are both high. As a result, the error of matching probability results in errors in estimating the transition probability between these states.

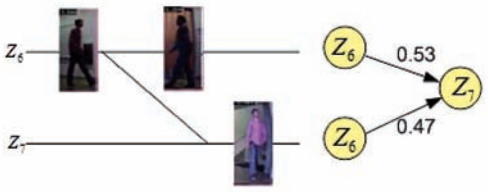
Figure 21(b) is another example. As our matching process relies on the color features, two subjects wearing clothes of similar colors (or the clothes have similar color distribution) may be matched incorrectly. However, the incorrect matching results sometimes may not affect the model estimation, as the example shown in Figure 21(c).



(a) One matching failure case. The new subject observed at state Z_3 is matched to itself at Z_4 and Z_{11}. As a result, the matching probability on the right hand is not estimated correctly.



(b) Another matching failure example. The new subject observed at state Z_{11} should be matched to the one at Z_6. However, an image of another subject observed at Z_{11} has higher similarity with that of the new subject. Therefore, the incorrect matching results affect the transition probability estimation.



(c) This example shows a matching failure which does not affect the model estimation. As two matching candidates are both from state Z_6, no matter which one has the higher similarity, it contributes to the transition probability from Z_6 to Z_7.

Fig. 21. Some failure cases in matching image observations to the subjects.

5. CONCLUSIONS

In this article, we propose a general probabilistic framework for traffic modeling and prediction with Bayesian inference, in which the transition probabilities between each pair of entry and exit states are modeled by a semi-Markov chain, and the traveling time durations between states are modeled by an exponential distribution. Subjects appearing in different poses are detected and matched via images acquired at different cameras, thereby facilitating estimation of the parameters in our model. We derive a maximum-likelihood estimator to the object association based on the observations. The proposed framework is validated with a camera sensor network in a smart building. Our experiments with more than 100,000 images show that the traffic patterns of humans in a building can be modeled and predicted well with our model.

REFERENCES

AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., AND CAYIRCI, E. 2002. A survey on sensor networks. *IEEE Commun. Mag. 40,* 8, 102–114.

CHEN, P., AHAMMAD, P., BOYER, C., HUANG, S.-I., LIN, L., LOBATON, E., MEINGAST, M., OH, S., WANG, S., YAN, P., YANG, A., YEO, C., CHANG, L.-C., TYGAR, D., AND SASTRY, S. 2008. CITRIC: A low-bandwidth wireless camera network platform. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. 1–10.

DALAL, N. AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 886–893.

DIAZ, I., HEIJLIGERS, M., KLEIHORST, R., AND DANILIN, A. 2007. An embedded low power high efficient object tracker for surveillance systems. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. 372–378.

DICK, A. R. AND BROOKS, M. J. 2004. A stochastic approach to tracking objects across multiple cameras. In *Proceedings of the Australian Joint Conference on Artificial Intelligence*. 160–170.

DOWNES, I., RAD, L., AND AGHAJAN, H. 2006. Development of a mote for wireless image sensor networks. In *Proceedings of the Conference on Cognitive Systems and Interactive Sensors*.

FENG, W., KAISER, E., FENG, W., AND BAILLIF, M. L. 2005. Panoptes: Scalable low-power video sensor networking technologies. *ACM Trans. Multimedia Comput. Commun. Appl. 1,* 2, 151–167.

FORSYTH, D. AND PONCE, J. 2002. *Computer Vision: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.

GILBERT, A. AND BOWDEN, R. 2006. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In *Proceedings of the European Conference on Computer Vision*. 125–136.

HARITAOGLU, I., HARWOOD, D., AND DAVIS, L. S. 2000. Event-based control for mobile sensor networks. *IEEE Trans. Pattern Anal. Mach. Intell. 22,* 8, 809–830.

HEATH, K. AND GUIBAS, L. 2007. Facenet: Tracking people and acquiring canonical face images in a wireless camera sensor network. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. 117–124.

HUANG, T. AND RUSSELL, S. 1997. Object identification in a Bayesian context. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*. 1276–1283.

INTEL CORPORATION. 2004. Intel PXA270 processor electrical, mechanical and thermal specification datasheet. Processor now manufactured by Marvell Technology Group Inc.

JAVED, O., RASHEED, Z., SHAFIQUE, K., AND SHAH, M. 2003. Tracking across multiple cameras with disjoint views. In *Proceedings of the IEEE International Conference on Computer Vision*. 952–957.

JAVED, O., SHAFIQUE, K., AND SHAH, M. 2005. Appearance modeling for tracking in multiple non-overlapping cameras. In *Proceedings of the IEEE Conference on Computer Vision*. 26–33.

KAMTHE, A., JIANG, L., DUDYS, M., AND CERPA, A. 2009. Scopes: Smart cameras object position estimation system. In *Proceedings of the European Conference on Wireless Sensor Networks*. 279–295.

KESHAVARZ, A., TABAR, A. M., AND AGHAJAN, H. 2006. Distributed vision-based reasoning smart home care. In *Proceedings of the 1st Workshop on Distributed Smart Cameras*.

KETTNAKER, V. AND ZABIH, R. 1999. Bayesian multi-camera surveillance. In *Proceedings of the IEEE Conference on Computer Vision*. 253–259.

KLEIHORST, R., ABBO, A., SCHUELER, B., AND DANILIN, A. 2007. Camera mote with a high-performance parallel processor for realtime frame-based video processing. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. 109–116.

KO, T., AHMADIAN, S., HICKS, J., RAHIMI, M., ESTRIN, D., AND SOATTO, S. 2010. Heartbeat of a nest: Using imagers as biological sensors. *ACM Trans. Sens. Netw. 6,* 3, 1–31.

KULKARNI, P., GANESAN, D., SHENOY, P., AND LU, Q. 2005. Senseye: A multi-tier camera sensor network. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*. 229–238.

LOBATON, E. J., AHAMMAD, P., AND SASTRY, S. 2009. Algebraic approach to recovering topological information in distributed camera networks. In *Proceedings of the International Confernece on Information Processing in Sensor Networks*. 193–204.

MAKRIS, D., ELLIS, T., AND BLACK, J. 2004. Bridging the gaps between cameras. In *Proceedings of the IEEE Conference on Computer Vision*. 205–210.

NIU, C. AND GRIMSON, E. 2006. Recovering non-overlapping network topology using far-field vehicle tracking. In *Proceedings of the IEEE International Conference on Pattern Recognition*. 944–949.

OMNIVISION TECHNOLOGIES INCORPORATED. 2006. OV9655 Color CMOS SXGA CAMERACHIP with OmniPixel technology datasheet. `http://www.ovt.com`.

PASULA, H., RUSSELL, S., OSTLAND, M., AND RITOV', Y. 1999. Tracking many objects with many sensors. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1160–1171.

RAHIMI, M., BAER, R., IROEZI, O. I., GARCIA, J. C., WARRIOR, J., ESTRIN, D., AND SRIVASTAVA, M. 2005. Cyclops: In situ image sensing and interpretation in wireless sensor networks. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*. 192–204.

RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. 2000. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision 40,* 2, 99–121.

SHUAI, Z., OH, S., AND YANG, M.-H. 2010. Traffic modeling and prediction using camera sensor networks. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*.

SONG, B. AND ROY-CHOWDHURY, A. K. 2008. Robust tracking in a camera network: A multi-objective optimization framework. *IEEE J. Sel. Top. Sign. Process. 2,* 4, 582–596.

SORO, S. AND HEINZELMAN, W. 2009. A survey of visual sensor networks. *Adv. Multimedia 2009*.

SUNDARRAJ, D., GIBBONS, P. B., AND PILLAI, P. S. 2006. Ensuring spatio-temporal consistency in distributed networks of smart cameras. In *Proceedings of the 1st Workshop on Distributed Smart Cameras*.

SWAIN, M. J. AND BALLARD, D. H. 1991. Color indexing. *Int. J. Comput. Vision 7,* 1, 11–32.

YAN, T., GANESAN, D., AND MANMATHA, R. 2008. Distributed image search in camera sensor networks. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*. 155–168.

YANG, D. B., GONZALEZ-BANOS, H. H., AND GUIBAS, L. J. 2003. Counting people in crowds with a real-time network of simple image sensors. In *Proceedings of the IEEE International Conference on Computer Vision*. 122–129.

Received February 2011; revised July, August 2011; accepted September 2011