Traffic Modeling and Prediction Using Camera Sensor Networks

Zaihong Shuai
EECS
University of California
Merced, CA 95344, USA
zshuai@ucmerced.edu

Songhwai Oh
EECS | ASRI
Seoul National University
Seoul, Korea
songhwai@snu.ac.kr

Ming-Hsuan Yang
EECS
University of California
Merced, CA 95344, USA
mhyang@ucmerced.edu

ABSTRACT

We propose a Bayesian 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 detection and matching of their images taken from cameras at different locations. Experiments with more than 2,500 images of 20 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.0 [**Computer Systems Organization**]: General; I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Camera Sensor Network, Traffic Modeling and Prediction, Pedestrian Detection, Image Matching, Smart Cameras

1. INTRODUCTION

In this paper, we propose a Bayesian framework for modeling traffic pattern of moving objects using information acquired from wireless sensor networks. The traffic pattern here refers to the moving pattern of humans, vehicles or other moving objects within the region of interest. We assume that the way objects move around within the network follows some regular patterns, as limited by the constraints of area layouts. Based on the observations, we extract useful information about how the objects move in the scenes. For example, we can predict the transition probability of an object moving from the sensing region of one sensor to another. In addition, we can estimate the expected traveling time for an object moving between regions using 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, i.e., we do not know the accurate positions or viewpoints of the sensors. The above-mentioned scenario requires 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 images taken from cameras with different field of views. The proposed framework can be applied, with different sensing devices, to other applications, such as surveillance, traffic monitoring, abnormality detection, location-based services, to name a few.

We conduct experiments in a smart building with a low-power, low-bandwidth distributed camera sensor network. With five CITRIC camera motes [2] placed at the intersections of stairways, hallways and elevators, we show that the traffic pattern of dwellers can be modeled and predicted well with the proposed model.

The contributions of our work are summarized as follows. We propose a Bayesian framework, based on semi-Markov process, for modeling the traffic patterns. The proposed approach deals with identity uncertainty, and hence it is applicable for realistic situations where a large number of objects move among the region of interest and their identities are not known *a priori*. Due to intrinsic characteristics of cameras, the association results are not guaranteed to be accurate, not to mention that the images from different camera are always under different viewpoint or lighting conditions. Thus, we derive a maximum-likelihood solution to exploit the association results in a probabilistic way. 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.

2. RELATED WORK

There is a rich literature on wireless sensor networks [1, 24] and a comprehensive review is beyond the scope of this

ICDSC 2010 August 31 – September 4, 2010, Atlanta, GA, USA

paper. In this section, we discuss the most relevant works in camera sensor networks, and their applications for modeling human activities.

There has been a consistent interest in applications with smart cameras, especially in tracking objects using multiple cameras [10, 23, 12, 9, 21, 13], object identification [11], learning network topology [19], people counting [28], etc. In [11], a traffic monitoring system is presented in which image matching and known traveling time are combined to establish vehicle correspondence between deployed camera sensors along a highway. However, it only models one single traffic pattern where the traffic generally follows highway lanes. Kettnaker and Zabih [15] introduce a Bayesian formalization to reconstruct the paths of objects across multiple cameras. While the cameras have non-overlapping field of view (FOV), they need to be calibrated. Their system requires a pre-defined set of allowable paths, transition probabilities and expected duration as a prior. Consequently, the proposed method has rather 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 [12]. 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 proposed [6] in which both Kalman filter and Markov model are used. The Kalman filter is used to resolve short tracks between frames, whereas the Markov model is applied to cope with discontinuity and track fast motion or motion 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 [13] 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 [9] incrementally updates transition matrix and color calibration mappings for tracking people across disjoint camera views. Song and Roy-Chowdhury [23] propose a stochastic, adaptive strategy for tracking multiple people in non-overlapping camera networks. With its long-term feature dependency models, their system can adaptively 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 be tracked within the view of each camera, which is a strong assumption.

In our experiment, a camera sensor network is formed using CITRIC camera motes [2]. Besides the CITRIC mote, there are numerous camera sensor platforms [22, 18, 8, 16, 7]. These platforms vary in configuration, processing capability, memory and image resolutions. In the last few years, there has been a growing interest in applications of camera sensor networks [17, 27, 25]. Yan et al. [27] implement a distributed image search system over a camera sensor network where each node is a search engine that senses, stores and searches for visual information. The node consists of a iMote2 mote [3] and low-power cameras with extended flash storage. Sundarraj et al. [25] propose an algorithm that matches images from multiple camera sensors, using spatial and temporal consistency. In [5], a real-time surveillance application for object tracking is proposed using WiCa [16] platform. Recently, Kamthe et al. [14] present a smart cameras object position estimation system using Cyclops [22] sensor network. Notwithstanding the demonstrated success in these applications, none of these applications develop efficient and effective data association algorithm to model the traffic pattern of moving objects.

What distinguishes our work from prior art is as follows. First, it is not necessary for our algorithm to track objects or to reconstruct their whole paths in sequences in order to analyze traffic patterns. Instead, the proposed algorithm entails only the local motion pattern of objects. Second, our framework is able to model the traveling time of moving objects. Furthermore, while our objective is not to track objects in the network, we can actually estimate the object paths probabilistically. We can also estimate the number of objects in the region of interest. With each mote reporting the number of objects entering/leaving the states (based on the human detection result) at any duration, the server can form a global view of the traffic flow.

3. PROBLEM FORMULATION

In this section, we present our framework for modeling and predicting traffic patterns. Our framework is generic and can be applied to numerous problems as we do not assume the specific sensing region or topology of sensors. It is also applicable 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 there are N sensors in the network, we denote $\mathcal{R}$ as the entire region of interest which covers the sensing area of all the sensors, i.e., $\{R_1, \cdots, R_N\} \subset \mathcal{R}$, where R_n is the sensing region of sensor n. They may be overlapped or not. Note that there exist regions uncovered by the sensors, so the union of $R_1, \ldots, R_N$ is a subset of $\mathcal{R}$. These sensing regions do not assume any particular structure in our formulation. As shown in Fig. 1, there are five regions in this sensor network where R_3 and R_4 are overlapped.

The possible entry/exit points within the whole sensing region are represented by S states $Z_1, \ldots, Z_S$, and usually $S \geq N$, as shown in Fig. 1, where the states are denoted by purple circles. In this example, each sensing region covers one or more states and one state may be covered by several sensors, e.g., Z_9. Assume a set of states, S_n, is covered by sensor n, then $\sum_{n=1}^{N} S_n = S + L$, where L is the total number of states that are covered by more than one sensor. In Fig. 1, there are 10 ($S = 10$) states and one ($L = 1$) of them is covered by more than one sensors.

With this formulation, the traffic pattern of interest refers to how objects travel from one state to another. We only consider the traffic pattern between R_i. The activity graph describes how objects move in the region of interest $\mathcal{R}$. Fig. 1(b) shows an example where each vertex represents a state 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 ($Z_1 \leftrightarrow Z_6$).

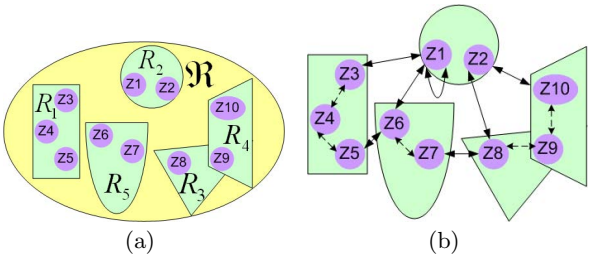


Figure 1: Sensing regions and corresponding activity graph. (a) The sensing region of each sensor may be overlapped or not. They can have different shape. (b) An activity graph. Z_1, Z_2 and Z_6, Z_7 are possible entrance/exit states in sensing region R_2 and R_5 respectively. These two regions are connected by a path ($Z_1 \leftrightarrow Z_6$). The graph representation distinguishes the U-turn and through traffic. The possible paths within and between R_i are represented by dotted and solid lines, respectively.

3.2 Mobility Model and Observation Model

We model the traffic pattern 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. Then the state transition is modeled as a Markov chain:

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

where i and j denote states Z_i and Z_j, respectively. However, unlike the conventional Markov chain, where the state transition happens instantaneously, we assume there is a delay at each transition. Let T_k be the traveling time between X_{k-1} and X_k and it has 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 our 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 times are accounted in this model. The initial state distribution is defined similarly to the conventional Markov chain. The semi-Markov chain describes the traffic pattern in the activity graph.

If an object is in R_n which covers Z_i from time t_{k-1} to t_k, 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 times. In our camera sensor network system, both images and traveling times are used as observations.

3.3 Learning With Known Identities

There are two sets of parameters we need to estimate in order to use our semi-Markov chain. They are state transition probabilities $\{p_{ij}\}$ and traveling time rates $\{\lambda_{ij}\}$.[1] In Section 3.2, we assume a single object and the parameter estimation is trivial since there is no uncertainty about object's identity. However, in a general setup, we need to consider the case with a large number of moving objects and the parameters cannot be estimated unless their identities are known. In the next two subsections, we describe how we can resolve this identity uncertainty (i.e., data association) problem and robustly estimate the parameters of the semi-Markov chain. In this section, we first assume the identities are known.

For ease of exposition, we show the method for estimating parameters for the outgoing transition from a single state Z_i. The parameters associated with other states can be estimated in a similar manner.

Now suppose that there are N_e objects that exited state Z_i. Then we can compute the likelihood of the outgoing 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 exited Z_i at time t_k is the same object that arrived at Z_j for the first time after t_k and $\gamma_{kj} = 0$, otherwise. If no object arrived at Z_j after time t_k, we also have $\gamma_{kj} = 0$. Once we know object identities, i.e., γ's, we can estimate the maximum likelihood of the transition probabilities by solving a constrained optimization problem[2] and obtain

$$\hat{p}_{ij} = \frac{\sum_{k=1}^{N_e} \gamma_{kj}}{\sum_{k=1}^{N_e} \sum_{j=1}^{S} \gamma_{kj}}. \tag{5}$$

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

$$\prod_{k=1}^{N_e} \prod_{j=1}^{S} (\lambda_{ij} \exp(-\lambda_{ij} t_{ij}))^{\gamma_{kj}}, \tag{6}$$

where t_{ij} is the traveling time when $\gamma_{kj} = 1$. The maximum likelihood estimate of the traveling time rate is

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

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 stated above. To address this problem, we need to first resolve the identity uncertainty.

3.4 Object Association

Using the observations from each sensor as an input, the object association process is to compute the matching probability of these 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 to make soft decisions. Let m be an object detected by sensor n covering state Z_i and let $Y_k^{n,m} = \{Y_t^{n,m} : t_{k-1} \le t \le t_k\}$ be the collection of measurements from the time the object m entered R_n (at time t_{k-1}) to the time the object exited (at time

[1] We also need to estimate the initial state distribution but it is ignored in this paper since its estimation is trivial.

[2] The constraint is the normalization property of the transition probabilities.

t_k). Without loss of generality, we assume that $Y_k^{n,m}$ are 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

$$\boldsymbol{q} = \{q_h\}_{h=1\ldots H}, \quad \sum_{h=1}^{H} q_h = 1, \tag{8}$$

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

Assume there are L collections of measurements from other sensors. They are listed as candidates to be compared with the measurement $Y_k^{n,m}$. For ease of exposition, $Y_k^{n,m}$ is simply denoted as Y_k and other L collections of measurements are denoted as $\{Y_l\}_{l=1\ldots L}$. The corresponding object of Y_k, as mentioned above, entered R_n from state Z_i. Measurements Y_l are selected as candidates since their corresponding objects exited from those states $\{Z_l\}$ that are possible previous-states of state Z_i, i.e., 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, i.e, how likely the object m is the candidate objects based on the measurements Y_k and $\{Y_l\}_{l=1\ldots L}$. Assume that the collection of measurements 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 histogram intersection algorithm [26]. Intuitively, the output 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 $\{s_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}} : t_{k-1} \leq t \leq t_k\}$, and we compute $d_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}}$, the distance between $\boldsymbol{q}_t^{n,m}$ and $\boldsymbol{\mu}_l$, as:

$$d_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}} = 1 - \frac{s_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}} - \min(W)}{\max(W) - \min(W)}. \tag{9}$$

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

$$p(m = l|\boldsymbol{q}_k^{n,m}) = \frac{\prod_{t=t_{k-1}}^{t_k} \exp(-d_{\boldsymbol{\mu}_l,\boldsymbol{q}_t^{n,m}})}{\sum_{l'=1}^{L} \prod_{t=t_{k-1}}^{t_k} \exp(-d_{\boldsymbol{\mu}_{l'},\boldsymbol{q}_t^{n,m}})}. \tag{10}$$

The computed probabilities are the association probabilities and we use them as approximations to $\mathbb{E}(\gamma_{li})$, i.e., the probability of an object leaving Z_l entering state Z_i.

3.5 Learning Under Identity Uncertainty

While we cannot directly solve for $\hat{p}_{ij}$ and $\hat{\lambda}_{ij}$ in (5) and (7) since we do not know the identities of objects, we can use the association probabilities computed above to resolve this issue.

Instead of maximizing the log likelihood to estimate the parameters, we maximize the expected complete log likelihood. The expected complete log likelihood for the transition probabilities are

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

which we can solve using the estimates found in the previous section. Then our estimates are

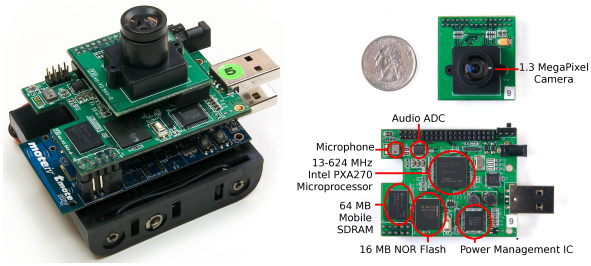
$$\hat{p}_{ij} = \frac{\sum_{k=1}^{N_e} \mathbb{E}(\gamma_{kj})}{\sum_{k=1}^{N_e}\sum_{j=1}^{S} \mathbb{E}(\gamma_{kj})}. \tag{11}$$

Similarly, the traveling times can be estimated as

$$\frac{1}{\hat{\lambda}_{ij}} = \frac{\sum_{k=1}^{N_e} \mathbb{E}(\gamma_{kj})t_{ij}}{\sum_{k=1}^{N_e} \mathbb{E}(\gamma_{kj})}. \tag{12}$$

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

4. EXPERIMENTS AND RESULTS



(a) (b)

Figure 2: CITRIC camera mote. (a) An assembled camera daughter board with Tmote Sky board. (b) A camera daughter board with major functional units outlined.

Our experiments are carried out using a network of CITRIC motes [2]. 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 Fig. 2). The CITRIC mote uses the OmniVision OV9655 CMOS image sensor [20] 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.

We carry out experiments in a smart building equipped with a network of CITRIC camera motes for modeling and predicting traffic patterns of dwellers. Fig. 3 shows the building layout and the placements of CITRIC motes. Five CITRIC motes are placed on two floors in a building at intersections of hallways as well as stairways, with four on the second floor of the building, and the other one on the first floor. Over 2,500 images of humans are collected by these motes. The training set consists of 1,442 images and the test set contains 1,136 images. In our experiment, the images are captured at a resolution of 320 × 240 pixels, which is sufficient for human detection. In the training phase, we

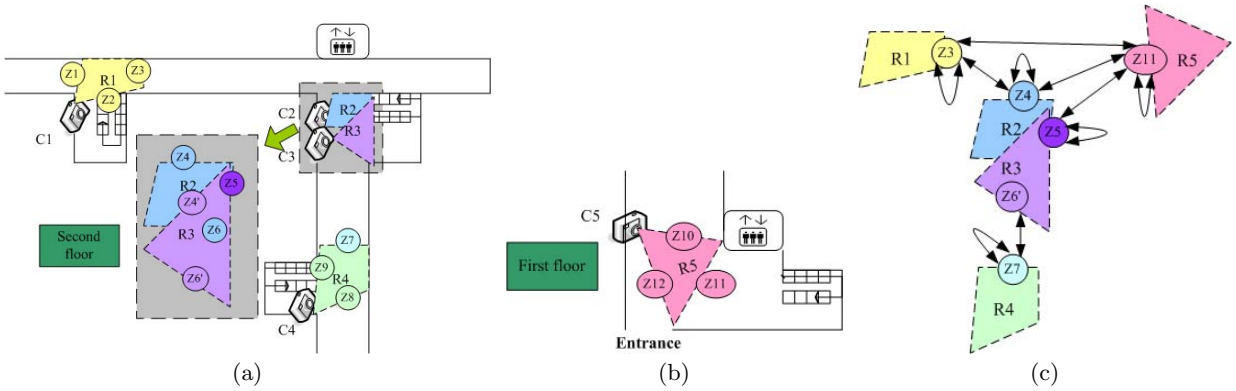


Figure 3: Experimental setup and activity graph. (a) Four camera motes are placed on the second floor of a building. The FOV (colored region) of each camera has different shape and Z_1 to Z_9 are possible entrance/exit points (states) of each region (R_2 and R_3 are shown in a larger region on the lower 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 at state Z_{11} can either take the elevator or the stairways 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). (c) Activity graph. Z_{11} is connected to Z_3, Z_4, and Z_5 as explained in (b) via different paths. The sink state Z_0 is not shown in the figure.

estimate the model parameters, i.e., $\{p_{ij}\}$ and $\{\lambda_{ij}\}$ as described in Section 3, using the images of dwellers detected from the motes. The observations used for human matching are the normalized RGB histograms of the upper-body part of the detected subjects. Once a subject is detected, its next state and expected arrival time can be predicted using the learned traffic model.

4.1 Training Phase

The FOVs of 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 Fig. 3(a)-(b) (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. 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, i.e., subjects walking through state Z_4 will definitely arrive at state Z_4'.[3] 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 Fig. 3(c) where 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 elevator or stairway. Likewise, Z_5 is connected to Z_{11} as a subject may take 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.

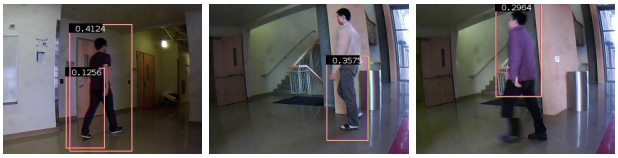
At each camera, two image sequences are collected with more than 20 people walking through this building. Each sequence lasts about 10 minutes (where images are acquired and saved at 4 frames per second), and all the raw images are transmitted to a central server for off-line training. Humans in these images are detected using a detector with Histogram of Oriented Gradient (HOG) descriptors [4] where the outputs are their image coordinates in the scenes.

Overall, the HOG-based detector performs well with our dataset with few false negatives and false positives. When one subject appears in the FOV of a camera, multiple frames of this subject will be captured by the camera. Consequently, even if a subject is not detected in some frames (i.e., false negatives), the negative effects 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 violate this rule are false positives and can be removed. Some other examples are shown in Fig. 4.

Furthermore, we can also remove some false positives based on temporal consistency. As we have continuous captured frames, so if at some frames the detection result (i.e., the coordinates of the bounding box) deviates from the results of other frames significantly, we can remove this frame, as either it is a false positive or there is another subject at that position (see Fig. 5). In both cases, the result from such frame can be removed without affecting learning the traffic pattern in our model.

Assume the camera motes are 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 are belonging to the same subject from all sequences

[3]That is, a subject entering state Z_4 will almost always arrive at state Z_4' (due to physical layout and sensing range), with negligible exceptions.



(a) (b) (c)

Figure 4: 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-coordinate of the bounding box is too large (i.e., the detected person is too small). (c) The y-coordinate of the bounding box is too small (i.e., the detected person is not on the floor).

(a) (b) (c)

Figure 5: Human detection results from three continuous 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).

acquired by one camera. 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 moving direction help in determining the entering state and leaving state of one subject, as the placements of cameras are approximately known. 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 Fig. 3(a), if the subject is observed to enter from left of the scene, then the entering state must be Z_1. If the subject is observed to leave from the far right end (with smaller bounding box), the leaving state is Z_3. Otherwise, if the bounding box is large and locates on the right side of the frame observed from C_1 when the subject leaves the scene, its leaving state is Z_2.

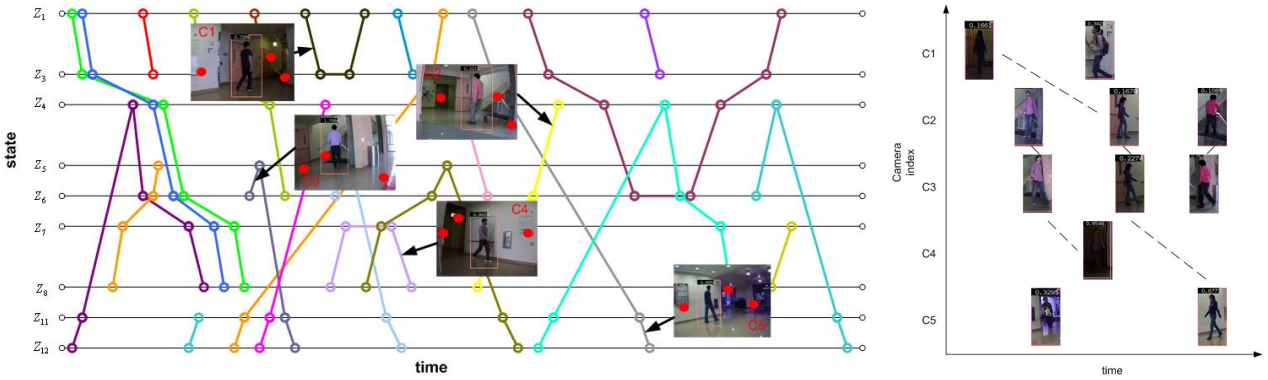
It is worth mentioning that human detection technique provides more useful information than methods using simple background subtraction with blob models. For example, multiple humans can be detected in a scene, thereby facilitating flow analysis of groups. Once a person is detected, the corresponding feature vector (i.e., Y of (3)) is extracted. In this work, normalized RGB color histogram is used as it is invariant to change in scale and viewpoint, thereby facilitating the matching process. In addition, we fit an ellipse within the bounding box of a detected human to remove background pixels. We have experimented with various representations and parameters, and find that the combination of normalized RGB histogram with 400 bins of a upper human torso performs best.

We exploit both spatial and temporal prior information for matching between clusters. In our formulation, a cluster is defined as the frames continuously captured by one camera and belongs to one subject. First, with prior spatial knowledge of camera placements and structure constraints, we know all the possible state transitions. For example, as seen in Fig. 3 (c), 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 (4)), i.e., 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 other 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 at time t_-, there is a detection by camera n. As mentioned above, we can infer the entering state of a subject, say, Z_i, from the coordinates of bounding box. Let the list of possible previous-states of Z_i be $E_i : \{Z_{i1}, \cdots, Z_{im}\}$, where m is the total number of possible state transitions end to 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 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 not empty, we first compute the distances between the image cluster at Z_i and other clusters in A_i. The histogram intersection method is used as it performs best in our experiments when compared to other metrics, e.g., Bhattacharyya distance, χ^2-distance, and sum of squared difference. If all the distances are relatively large, the subject is regarded as a new person appearing from some blind spots, i.e., entering the scene from Z_0. Otherwise, the corresponding matching probability is computed using (10). If A_i is empty, it means there are no other suitable image clusters to compare with, and the subject is also regarded as new. For each cluster, if there exists no other clusters to choose it as matching candidate, the corresponding subject is considered as disappeared in the scene, i.e., arriving at state Z_0.

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 probabilities and traveling times can be estimated as described in Section 3.

Fig. 6(a) shows some example sequences used in the training phase where each trajectory describes one possible path. Note that not all the states are shown in the figure, as some of them do not contribute to the traffic model, e.g., Z_2 and Z_9 (where subjects enter into regions not monitored by the cameras). These paths indicate that subjects move freely in various patterns. It is worth noting that these trajectories can be identified and matched through the images acquired at different cameras using our algorithm. Fig. 6(b) shows 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



(a) (b)

Figure 6: (a) Sample sequences used in our experiment. 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 stand for the states. Sample images acquired at five cameras are also shown next to the states (best viewed on a high-resolution LCD display). (b) Some detection results from image frames captured by CITRIC motes. The x-axis and y-axis represent the time and camera index.

same subject may change dramatically as viewed by different cameras, due to variation of lighting and response of CCD sensors.

Experimental results using the training set are shown in the second column of Table 1 which lists the estimated probability ($p_{i\to j}$) with duration time ($t_{i\to j}$, i.e., $\frac{1}{\lambda_{i\to j}}$, same as in (12)) in parenthesis of training phase, while the third column presents the corresponding ground truth values. The ground truth values are obtained by visually matching all the frames, and counting the frequency of how the subjects move between states. Overall, these estimated probabilities and traveling time of our model match the ground truth values well. Compared with the ground truth, the average error of all state transition probabilities is 0.0556 and the standard deviation is 0.08.

There are a few cases that our model does not estimate state transition probabilities well. For example, from the ground truth data we know there is no subject moving from Z_{11} to Z_{11}, but the estimated probability of moving from Z_{11} to Z_{11} is 0.1181 with an average traveling time of 26.44 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 Fig. 3, two adjacent cameras, C_2 and C_3, have overlapped FOVs, and thus most subjects appearing in 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 false matching by exploiting such additional cues.

Table 1: Estimated parameters and ground truth

State transition	Training	Ground truth	Test
$p_{3\to0}$ ($t_{3\to0}$)	0.3482 (–)	0.4 (–)	0.3333(–)
$p_{3\to3}$ ($t_{3\to3}$)	0.1170 (21.01)	0.1 (22)	0.1111 (28)
$p_{3\to4}$ ($t_{3\to4}$)	0.4023 (32.54)	0.4 (35)	0.4444 (32.75)
$p_{3\to11}$ ($t_{3\to11}$)	0.1325 (41.93)	0.1 (75)	0.1111 (91)
$p_{4\to0}$ ($t_{4\to0}$)	0.3241 (–)	0.4 (–)	0.4444 (–)
$p_{4\to3}$ ($t_{4\to3}$)	0.3681 (20.67)	0.3 (18)	0.3333 (31.33)
$p_{4\to4}$ ($t_{4\to4}$)	0.0013 (25.98)	0 (–)	0 (–)
$p_{4\to11}$ ($t_{4\to11}$)	0.3066 (54.13)	0.3 (51)	0.2222 (48)
$p_{5\to0}$ ($t_{5\to0}$)	0.2778 (–)	0.3 (–)	0.3 (–)
$p_{5\to5}$ ($t_{5\to5}$)	0.1397 (12.94)	0.1 (12)	0.2 (22.5)
$p_{5\to11}$ ($t_{5\to11}$)	0.5825 (24.62)	0.6 (29)	0.5 (26.8)
$p_{6\to0}$ ($t_{6\to0}$)	0.2760 (–)	0.5 (–)	0.5714 (–)
$p_{6\to6}$ ($t_{6\to6}$)	0.2683 (17.63)	0.1 (22)	0.1429 (20)
$p_{6\to7}$ ($t_{6\to7}$)	0.4557 (18.49)	0.4 (15.5)	0.2857 (21.5)
$p_{7\to0}$ ($t_{7\to0}$)	0.1860 (–)	0.1818 (–)	0.1429 (–)
$p_{7\to6}$ ($t_{7\to6}$)	0.4701 (13.89)	0.5455 (11.25)	0.5714 (28)
$p_{7\to7}$ ($t_{7\to7}$)	0.3439 (17.84)	0.2727 (20)	0.2857 (22.5)
$p_{11\to0}$ ($t_{11\to0}$)	0.2664 (–)	0.25 (–)	0.25 (–)
$p_{11\to3}$ ($t_{11\to3}$)	0.1812 (53.78)	0.1667 (57.5)	0.125 (83)
$p_{11\to4}$ ($t_{11\to4}$)	0.2348 (33.96)	0.25 (24)	0.25 (39.5)
$p_{11\to5}$ ($t_{11\to5}$)	0.1994 (24.33)	0.3333 (22.25)	0.375 (30.67)
$p_{11\to11}$ ($t_{11\to11}$)	0.1181 (26.44)	0 (–)	0 (–)

4.2 Test Phase

We compare our parameter estimation results with the ground truth of test sequences which is obtained by visually inspecting the trajectories of all the subjects. The results of the test sequences are shown in the fourth column of Table 1. As evident in the table, our model is able to learn the transition probabilities well. The ground truth that we gather from the training set is assumed to be representative (which is assumed by almost all statistical learning frameworks) as long as the number of data points is sufficiently large. The errors of estimated state transition probabilities and traveling times using the training set are consistency smaller than those using the test set. These results indicate that the learned model parameters do not overfit the training data. More results with discussions can be found on our web page `http://eng.ucmerced.edu/people/zshuai/icdsc10.html`.

4.3 Discussion

As described earlier, multiple subjects can be detected with the HOG-based human detection algorithm. This is in contrast to prior work where tracking algorithms are used. 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. In addition, we plan

to carry out large-scale experiments (e.g., analyzing the traffic patterns using the data collected throughout one week or one month).

5. CONCLUSIONS

In this paper, we propose a general framework for traffic modeling and prediction with Bayesian inference, where the transition probabilities and the traveling time durations between states are modeled by semi-Markov chain. Subjects appearing in different pose 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 for the case with identity uncertainty, making the proposed method more suitable for realistic situations. The proposed framework is validated with a camera sensor network in a smart building. With five cameras placed at intersection of stairways, elevators, and hallways, our experiments with more than 2,500 images show that the traffic patterns of dwellers can be modeled and predicted well with our model.

6. ACKNOWLEDGMENTS

Z. Shuai and S. Oh are supported in part by the Army Research Office MURI grant W911NF-06-1-0076. M.-H. Yang is supported in part by a Google Faculty Award.

7. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communication Magazine*, 40(8):102–114, 2002.

[2] P. Chen, P. Ahammad, C. Boyer, S.-I. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, A. Yang, C. Yeo, L.-C. Chang, D. Tygar, and S. Sastry. CITRIC: A low-bandwidth wireless camera network platform. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–10, 2008.

[3] Crossbow Technology. Imote2 ipr2400 datasheet. `http://www.xbow.jp/imote2.pdf`.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[5] I. Diaz, M. Heijligers, R. Kleihorst, and A. Danilin. An embedded low power high efficient object tracker for surveillance systems. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, pages 372–378, 2007.

[6] A. R. Dick and M. J. Brooks. A stochastic approach to tracking objects across multiple cameras. In *Proceedings of Australian Joint Conference on Artificial Intelligence*, pages 160–170, 2004.

[7] I. Downes, L. Rad, and H. Aghajan. Development of a mote for wireless image sensor networks. In *Proceedings of Cognitive Systems and Interactive Sensors*, 2006.

[8] W. Feng, E. Kaiser, W. Feng, and M. L. Baillif. Panoptes: scalable low-power video sensor networking technologies. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 1(2):151–167, 2005.

[9] A. Gilbert and R. Bowden. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In *Proceedings of European Conference on Computer Vision*, pages 125–136, 2006.

[10] I. Haritaoglu, D. Harwood, and L. S. Davis. Event-based control for mobile sensor networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[11] T. Huang and S. Russell. Object identification in a Bayesian context. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 1276–1283, 1997.

[12] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *Proceedings of IEEE International Conference on Computer Vision*, pages 952–957, 2003.

[13] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *Proceedings of IEEE Conference on Computer Vision*, pages 26–33, 2005.

[14] A. Kamthe, L. Jiang, M. Dudys, and A. Cerpa. Scopes: Smart cameras object position estimation system. In *European Conference on Wireless Sensor Networks*, pages 279–295, 2009.

[15] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Proceedings of IEEE Conference on Computer Vision*, pages 253–259, 1999.

[16] R. Kleihorst, A. Abbo, B. Schueler, and A. Danilin. 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*, pages 109–116, 2007.

[17] T. Ko, S. Ahmadian, J. Hicks, M. Rahimi, D. Estrin, and S. Soatto. Heartbeat of a nest: Using imagers as biological sensors. *ACM Transactions on Sensor Networks*, 6(3):1–31, 2010.

[18] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu. Senseye: A multi-tier camera sensor network. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 229–238, 2005.

[19] C. Niu and E. Grimson. Recovering non-overlapping network topology using far-field vehicle tracking. In *IEEE International Conference on Pattern Recognition*, pages 944–949, 2006.

[20] Omnivision Technologies Incorporated. *OV9655 Color CMOS SXGA CAMERACHIP with OmniPixel Technology Datasheet*, 2006. `http://www.ovt.com`.

[21] H. Pasula, S. Russell, M. Ostland, and Y. Ritov'. Tracking many objects with many sensors. In *International Joint Conference on Artificial Intelligence*, pages 1160–1171, 1999.

[22] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor networks. In *ACM Conference on Embedded Networked Sensor Systems*, pages 192–204, 2005.

[23] B. Song and A. K. Roy-Chowdhury. Robust tracking in a camera network: A multi-objective optimization framework. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):582–596, 2008.

[24] S. Soro and W. Heinzelman. A survey of visual sensor networks. *Advances in Multimedia*, 2009.

[25] D. Sundarraj, P. B. Gibbons, and P. S. Pillai. Ensuring spatio-temporal consistency in distributed networks of smart cameras. In *Proceedings of the First Workshop on Distributed Smart Cameras*, 2006.

[26] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[27] T. Yan, D. Ganesan, and R. Manmatha. Distributed image search in camera sensor networks. In *ACM Conference on Embedded Networked Sensor Systems*, pages 155–168, 2008.

[28] D. B. Yang, H. H. Gonzalez-banos, and L. J. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *Proceedings of IEEE International Conference on Computer Vision*, pages 122–129, 2003.