Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2011, Article ID 176026, 25 pages
doi:10.1155/2011/176026

Research Article

Efficient Data Association in Visual Sensor Networks with Missing Detection

Jiuqing Wan and Qingyun Liu

Department of Automation, Beijing University of Aeronautics and Astronautics, Beijing 100191, China

Correspondence should be addressed to Jiuqing Wan, wanjiuqing@gmail.com

Received 26 October 2010; Revised 16 January 2011; Accepted 18 February 2011

Academic Editor: M. Greco

One of the fundamental requirements for visual surveillance with Visual Sensor Networks (VSN) is the correct association of camera's observations with the tracks of objects under tracking. In this paper, we model the data association in VSN as an inference problem on dynamic Bayesian networks (DBN) and investigate the key problems for efficient data association in case of missing detection. Firstly, to deal with the problem of missing detection, we introduce a set of random variables, namely routine variables, into the DBN model to describe the uncertainty in the path taken by the moving objects and propose the high-order spatio-temporal model based inference algorithm. Secondly, for the problem of computational intractability of exact inference, we derive two approximate inference algorithms by factorizing the belief state based on the marginal and conditional independence assumptions. Thirdly, we incorporate the inference algorithm into EM framework to make the algorithm suitable for the case when object appearance parameters are unknown. Simulation and experimental results demonstrate the effect of the proposed methods.

1. Introduction

Consisting of a large number of cameras with nonoverlapping field of view, Visual Senor Networks (VSNs) have been frequently used for surveillance of public locations such as airports, subway stations, busy streets, and public buildings. The visual nodes in VSN are not working independently; instead, they can transmit information to a processing centre or communicate with each other. Typically, in the region covered by the VSN there are several moving objects (persons, cars, etc.), presenting in one camera at a certain time and reappearing in another after a certain period. The visual information captured by VSN can be used for interpreting and understanding the activities of moving objects in the monitored region. One of the basic requirements for achieving these goals is to accurately associate the observations produced by the visual node with the track of each object of interest. It is interesting to note that a similar problem also arise, in the multitargets tracking (MTT) research, where the goal is to associate the several distinct track segments produced by the same target. For example, Yeom et al. [1] proposed a track segments association technique for improving the track continuity in airborne early warning system using discrete optimization on the possible matching pairs of track segments given by forward prediction and backward retrodiction. However, the target motion model used in multitargets tracking is not available in VSN, as large blind regions always exist between camera nodes.

Appearance information can be used to associate the observation with object's track, provided the characteristics of the object appearance are known or have been learnt. However, the appearance observations of the same object generated by different visual nodes may vary dramatically due to changes in the illumination of scene or the photometric property of cameras. Despite the huge amount of effort to overcome these difficulties using intercamera color calibration [2] or appearance similarity models [3], the association performance based solely on appearance cues is still unsatisfactory. On the other hand, the spatiotemporal observations such as the time of object visiting the specific camera and the location of that camera, combined with the structural knowledge of monitored region, can be used to improve the accuracy of data association. Representing the prior knowledge of the object's characteristics and the monitored region by a graphical probabilistic model, the data

association problem can be solved by Bayesian inference [4–8].

However, the introduction of spatiotemporal information greatly complicates the association problem in the following two aspects. First, as the spatiotemporal observations of the same object from cameras in the VSN are inter-dependent, and the number of the association hypothesis usually increases exponentially with the accumulation of observations, rendering the exact inference algorithm intractable. In fact, intractability is an intrinsic property of the data association problems, no matter in VSN or in traditional multitargets tracking [9]. In traditional MTT community, data association problem can be solved by approximate algorithms such as Multiple Hypothesis Tracking (MHT) [10], Probabilistic Multiple Hypothesis Tracking (PMHT) [11], and Joint Probability Data Association (JPDA) [12]. However, the assumption of motion smoothness in traditional MTT is not available in VSN. Second, the performance of the spatiotemporal observation-based association algorithms is more vulnerable than that of the appearance-based methods to the unreliable detection, including false measurement and missing detection. In practice, unreliable detection is difficult to avoid due to the bad observation condition or the occlusion of the object of interest. The problem of false measurement can be alleviated by deleting the observations with low likelihood. However, missing detection is more difficult to handle as it is not easy to know whether an object is miss detected based on the information on a single camera. Moreover, missing detection can result in very low posterior probability of the true object trajectory, as most spatiotemporal model-based inference algorithms rely on the assumption that the object can be detected reliably. Therefore, in this paper we focus our attention on the problem of missing detection and assume that there is no false or clutter measurement.

In fact, unreliable detection may also be encountered in traditional MTT applications such as low elevation sea surface target (LESST) tracking, where the SNR at receiver can be dramatically reduced due to the presence of multipath fading. For example, Wang and Mušicki [13] present a series of integrated PDA type filters which can output not only target state estimate but also a measure of track quality, taking into account the existence of target and the SNR of sensor. Godsill and Vermaak [14] deal with the problem of unreliable detection by incorporating a new observation model based upon a Poisson assumption for both targets and clutter into the variable rate particle filter framework.

In this paper, we present a novel method for data association in VSN based jointly on appearance and spatiotemporal observation, overcoming the difficulties mentioned above. After a brief review of the related works, in Section 3 we model the data association problem with dynamic Bayesian networks, where a set of routing variables are introduced to overcome the problem of missing detection. In Section 4 we present the forward and backward exact inference algorithms for data association in DBN and show their intractability when the number of objects grows. To reduce the computational burden, in Section 5 we derive two kinds of approximation inference algorithms by factorizing the joint probability distribution based on different independence assumptions. To apply the algorithms when objects appearance model is unavailable, in Section 6 we incorporate the proposed inference algorithms into EM framework, where the data association and parameter estimation problems are solved simultaneously. Simulation and experimental results are presented in Section 7 and conclusions are given in Section 8.

2. Related Works

The data association in VSN can be considered as the process of partitioning the set of observations collected by all cameras in VSN into several exhaustive and exclusive subsets, such that the observations belonging to each subset are believed to come from a single object. Then the data association problem can be solved by finding the partition with the highest posterior probability. Usually, the joint probability of partitions and observations is encoded by some graphical model. Pasula et al. [4] proposed a graphical model to represent the probabilistic relationships between the assignments variables, observations, and the hidden intrinsic and dynamic states of the objects under tracking. The introduction of hidden states in [4] avoids the combinatorial explosion in the number of the model parameters. Kim et al. [7] provided a first-order Markov model describing the activity topology of the camera networks, with so-called super nodes of the ordered entry-exit zone pairs and directional edges weighted by the likelihood of transition between cameras and the travel time. The model is superior in distinguishing traffic patterns compared with conventional activity topology models. Zajdel and Kröse [6] used dynamic Bayesian networks (DBNs) as generative model of observations from a single object. Every partition of the entire observations translates into a specific structure of a larger network that comprised multiple disconnected DBNs. The authors provided an EM-like approach to select the most likely structure and learn the model parameters. In the works mentioned above, although the association performance has been studied as a function of the increasing observation noise, none of them considered the problem of missing detection explicitly in their models. Van De Camp et al. [8] modeled the behavior of a single person by a Hidden Markov Model (HMM), with the hidden state representing the location of the person under tracking. In [8], each camera was represented by two states to be able to model the case of a person passing a camera without being detected.

In the above works, the complexity nature of data association reflects in the intractability of the partition space, which expands combinatorially with the number of observations. In [4, 7], the authors resort to Markov Chain Monte Carlo (MCMC) sampling method to represent the partition space by a set of samples with high posterior probability. Although MCMC-based method has been widely used in data association [15] and object tracking [16] problems and is simple to implement, it is usually computational intensive and rather sensitive to initial samples. In [6], the authors

approximate the full partition space by a Multiple Hypothesis Tracker- (MHT-) like approach, preserving the several most likely partitions and extending each partition with the subsequent observations. However, it is questionable if the true partition is also discarded as unlikely ones by a simple threshold value.

An alternative way to solve data association problem in VSN is to assume an imaginary label for each observation, indicating which object it comes from. As the label cannot be observed, it is treated as a hidden random variable. By inferring the posterior distribution of the imaginary label based on all available evidences, the object corresponding to each observation can be determined without explicit enumeration of the partitions of observations. In [5], the imaginary label is identified by probabilistic clustering the observations with an extension of Gaussian Mixture Model (GMM), where a set of hidden pointer variables are introduced to capture Markov dependencies between spatiotemporal observations generated by the same Gaussian kernel. However, the state space of the auxiliary hidden variables grows exponentially with the number of objects. This makes it very difficult to marginalize these variables out. The author solves the problem by Assumed Density Filtering (ADF) algorithm [17], where the joint distribution is replaced with a factorial approximation. Following the same way, in [18] the author presents a hybrid graphical model with discrete states identifying objects labels and continuous states underlying appearance and spatiotemporal observations. The model parameters are treated as a part of the state, allowing them to be learnt automatically with Bayesian inference. However, the inference is still difficult in that the posterior joint distribution take the form of mixtures of an intractable number of components due to the inherent association ambiguity.

In our work we also use the auxiliary pointer variables in [5, 18] to indicate the last observation of each object directly before the current one, but our work is differentiated from them in the following two aspects. First, the model in [5, 18] is based on the assumption that the objects cannot be miss detected by cameras. If this assumption is violated, as is often the case in practice, the association accuracy of the algorithm decreases significantly. In our work we tackle this problem by introducing another set of hidden variables indicating the path taken by the object from one camera to another. By considering all possible paths with limited length between camera nodes, the robustness of the algorithm against missing detection is greatly improved. Second, in [5, 18] the author factorized the joint distribution into the product of distributions of the label variable and single pointer variable to avoid the combinatorial explosion of state space. However, as the Markov transition process of the pointer variable is deterministic, the mixing rate of the process is zero. Theoretically, for this case the accumulated approximation error bound is infinite [17]. In contrast, we propose another scheme of factorization of the joint distribution based on the conditional independence between the pointer variables given the imaginary label. The proposed approximate inference demonstrates better association performance in simulation and experiment.

There are also other ways to solve the data association problems in VSN. It is very interesting to note that Loy et al. [19] proposes a novel approach for modeling correlations between activities observed by multiple nonoverlapping cameras. After decomposing each camera view into semantic regions according to the similarity of local activity patterns, the correlation structure of the semantic regions is discovered automatically by Cross Canonical Correlation Analysis. The resulting correlation structure can be used to improve data association performance by reducing the searching space and resolving the ambiguities arisen from similar visual features presented by different objects. Song and Roy-Chowdhury [20] propose a multiobjective optimization framework combining short-term feature correspondences across the cameras with long-term feature dependency models. The overall solution strategy involves adapting the similarities between features observed at different cameras based on the longterm models and finding the stochastically optimal path for each object. Based on activity topology graph, Kettneker and Zabih [21] transform the multicamera tracking problem into a linear assignment problem, which can be solved in polynomial time. However, since the weighted assignment algorithm uses correspondences between only two observations, other useful information such as the length and the frequency of path should be decomposed into "between-two-cameras" terms with a decomposable assumption. A high-order transition model can be used to associate the observations [22], but it turns the problem into multidimensional assignment problem.

3. Bayesian Modeling

In this section we formulate the problem of data association in VSN with missing detections and show that it can be solved by inference on dynamic Bayesian networks. Suppose that K objects are moving in the region monitored by M cameras, as shown in Figure 1. We use $A = \{a_{uv}\}_{u,v=1}^{M}$ to denote the parameter matrix of the VSN, each element of A consists of three components, that is, $a_{uv} = (\pi_{uv}, t_{uv}, s_{uv})$, π_{uv} is the transition probability of object moving from camera u to camera v, and $\pi_{uv} = 0$ means there is no edge between camera u and v. t_{uv} and s_{uv} are the mean and variance of the traveling time between u and v, respectively. Since we focus on camera-to-camera trajectory, we do not analyze the maneuvers of an object within the FOV of a single camera. The duration of object's presence in a viewing field is assumed to be significantly shorter than the travel times between cameras. Therefore, we will represent the interval within a camera field as a single timestamp and derive a "virtual" observation $y_i = \{o_i, d_i, c_i\}$, automatically or manually, from the sequence of frame captured by the camera once an object passed by. Here, o_i is the measurement of object appearance characteristics, and it can be the average of measurements on different frames, or just the measurement on a single fame; d_i is the time when observation was made, and it can be the time instant of object entry or departure, or the median of them; c_i is the camera at which the observation was made. All the generated

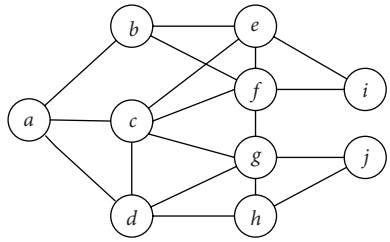

Figure 1: Topology of visual sensor networks. Circles depict cameras; edges depict path between cameras.

observations are collected to a data processing center and reordered according to their generating time, that is, $d_i < d_j$ if $i < j$ for any two observations y_i and y_j.

For each observation we introduce a labeling random variable $x_i \in \{1,\ldots,K\}$; $x_i = k$ indicates that the observation y_i is originated from the object k. In addition, we introduce another set of auxiliary random variables $z_i = \{z_i^{(k)}\}_{k=1}^{K}$, each $z_i^{(k)} \in \{0,\ldots,i-1\}$ indicates which of the observations $y_0,\ldots,y_{i-1}$ was the last observation of object k directly before the observation y_i, and $z_i^{(k)} = 0$ means y_i is the first observation of object k. Both x_i and z_i are unobserved and considered as hidden states to be estimated based on available observations. The goal of data association is to calculate the marginal posterior distribution of x_i, that is, $p(x_i \mid y_{0:i})$. In this section we first define the state transition model and observation model for the case of reliable detection and then introduce the routing random variables to accommodate the missing detections; finally we express the generating process of the observations sequence compactly with dynamic Bayesian networks.

3.1. State Transition Model. Based on the definition of hidden state variable x_i and z_i, it is reasonable to assume that the state evolve as a first-order Markov process. The state transition model can be written as

$$p(x_i, z_i \mid x_{i-1}, z_{i-1}) = p(x_i) f\left(z_i \mid x_{i-1} = k, z_{i-1}^{(k)} = l\right). \quad (1)$$

The prior probability $p(x_i)$ can be assumed to follow a uniform distribution if no prior knowledge about x_i is available. It should be noticed in above model that if x_{i-1} and z_{i-1} are given, the value of z_i is determined. Specifically, if the observation y_{i-1} is produced by object k, that is, $x_{i-1} = k$, then $z_i^{(k)}$ takes value of $i-1$ and other components in z_i remain unchanged, that is,

$$z_i^{(k)} = z_{i-1}^{(k)}[x_{i-1} \neq k] + (i-1)[x_{i-1} = k], \quad (2)$$

where $[g] \equiv 1(0)$ if and only if the logical variable g is true (false).

3.2. Observation Model. The observation includes appearance measurement and spatiotemporal measurement. We assume that they are conditionally independent given the current state, and both of them follow Gaussian distribution. The appearance observation model of a given object is

$$p(o_i \mid x_i = k) = N\left(o_i; \mu_k, \sigma_k^2\right), \quad (3)$$

where μ_k and σ_k^2 are mean and variance of the appearance of the kth object. The appearance observation is independent of the state z_i. The spatiotemporal observation is dependent on x_i, z_i and past observations $y_{0:i-1}$, as follows:

$$\begin{aligned} &p\left(d_i, c_i \mid x_i = k,\ z_i^{(k)} = l, y_{0:i-1}\right) \\ &\quad = p\left(d_i \mid x_i = k,\ z_i^{(k)} = l,\ d_l, c_l = u,\ c_i = v\right) \\ &\qquad \times p\left(c_i = v \mid x_i = k,\ z_i^{(k)} = l,\ d_l, c_l = u\right) \\ &\quad = \begin{cases} c, & l = 0, \\ \pi_{uv} N(d_i - d_l; t_{uv}, s_{uv}), & l \neq 0. \end{cases} \end{aligned} \quad (4)$$

Note that the spatiotemporal observation only depends on $z_i^{(k)}$ if $x_i = k$. As the observation y_0 is undefined, we set the likelihood in the case of $l = 0$ to a constant value c.

3.3. Missing Detection. At each monitoring camera, missing detection is unavoidable due to the unfavourable observing conditions. When the object of interest is miss detected, the true trajectory of that object cannot be expressed in terms of any sequence of state variable $z_i^{(k)}$, $i = 1,\ldots,N$. This will introduce unpredictable errors in the likelihood evaluation according to (4) and hence deteriorate the performance of data association algorithm significantly. To deal with this problem, we introduce another set of random variable, namely, routing variables $\omega_i = \{\omega_i^{(u,v)}\}_{u,v=1}^{M}$, to describe the uncertainty in the object moving path. The routing variable $\omega_i^{(u,v)}$ indicates the path with maximum length of L taken by an object moving from camera u to v. It is a discrete random variable taking values in the set $\{1,\ldots,R_{uv}^L\}$, where R_{uv}^L is the number of path form u to v not longer than L. The path length here is the number of camera nodes between u and v; $L = 0$ means that u and v are connected directly. The choice of L depends on the rate of missing detection, larger L for a higher missing detection rate, and vice versa. ω_i is a very large set of variables as it enumerates all pairing of cameras in the VSN. It seems that this will bring a huge computational burden in the inference computation. Fortunately, it turns out in Section 4 that most of the routing variables can be summed out during inference and the introduction of routing variable increases the computational burden very slightly.

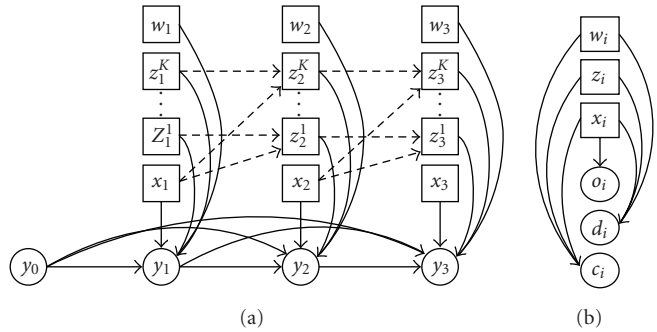

Figure 2: (a): Dynamic Bayesian networks model; (b) dependency in a single time slice. Solid arrows depict stochastic dependency; dashed arrows depicted deterministic one. Squares depict discrete random variables; circles depicted continuous ones.

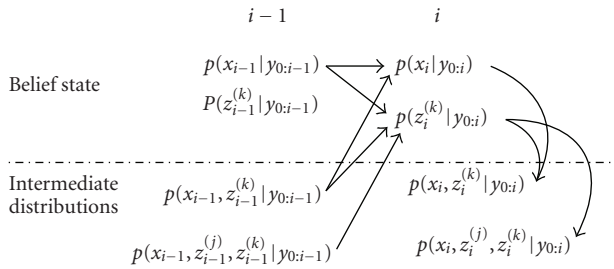

Figure 3: Belief state propagation in forward pass of *Approximate inference* I. We only need to maintain the belief state the intermediate distributions can be calculated based on the independence assumptions when necessary, as indicated by the arrows within each time slice.

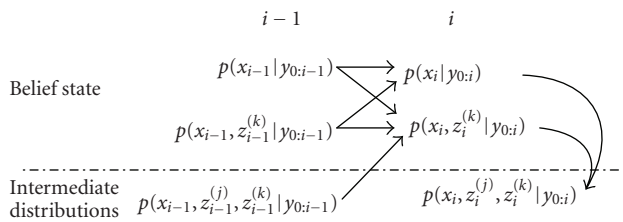

Figure 4: Belief state propagation in forward pass of *Approximate inference* II. We only need to maintain the belief state the intermediate distributions can be calculated based on the independence assumptions when necessary, as indicated by the arrows within each time slice.

Treating (x_i, z_i, ω_i) as hidden state, the state transition model can be written as

$$\begin{aligned} &p(x_i, z_i, \omega_i \mid x_{i-1}, z_{i-1}, \omega_{i-1}) \\ &\quad = p(x_i)p(\omega_i)f\left(z_i \mid x_{i-1} = k, z_{i-1}^{(k)} = l\right). \end{aligned} \tag{5}$$

Note that z_i is independent of ω_{i-1} given x_{i-1} and z_{i-1}, and x_i, z_i and ω_i are assumed to be mutually independent. When there is no observation to be conditioned on, the prior probability of ω_i is determined by the topological structure of the camera networks. So it is reasonable to assume that the random variable ω_i is independent of x_i and z_i. However, when ω_i is conditioned on $y_{0:i}$, it is dependent on x_i and z_i through the spatiotemporal model (7). The prior probability of object moving path $p(\omega_i)$ can be calculated according to the transition probabilities along that path. We use $\omega_i^{(u,v)} = (u, w_0^{(r)}, \ldots, w_{L-1}^{(r)}, v)$ to denote the rth path of length L from u to v, where $w^{(r)}$ is the intermediate nodes. Then the prior probability of object taking the rth path from u to v is

$$\overline{\pi}_{uv}^{(r)} \triangleq p\left(\omega_i^{(u,v)} = r\right) = \frac{\pi_{uw_0^{(r)}}\left(\prod_{l=1}^{L-1}\pi_{w_{l-1}^{(r)}w_l^{(r)}}\right)\pi_{w_{L-1}^{(r)}u}}{\sum_r \pi_{uw_0^{(r)}}\left(\prod_{l=1}^{L-1}\pi_{w_{l-1}^{(r)}w_l^{(r)}}\right)\pi_{w_{L-1}^{(r)}u}}. \tag{6}$$

The spatiotemporal observation model changed to

$$\begin{aligned} &p(d_i, c_i \mid x_i = k,\ z_i, \omega_i, y_{0:i-1}) \\ &\quad = p\left(d_i \mid x_i = k, z_i^{(k)} = l,\ \omega_i^{(u,v)} = r,\ d_l, c_l = u,\ c_i = v\right) \\ &\qquad \times p\left(c_i = v \mid x_i = k, z_i^{(k)} = l,\ \omega_i^{(u,v)} = r,\ d_l, c_l = u\right) \\ &\quad = \begin{cases} c, & l = 0, \\ N\left(d_i - d_l; \overline{t}_{uv}^{(r)}, \overline{s}_{uv}^{(r)}\right), & l \neq 0. \end{cases} \end{aligned} \tag{7}$$

Based on the Gaussian assumption, the mean and variance parameters in (7) can be calculated directly from the parameter matrix A of the VSN. The mean time of the object moving from u to v along path r is

$$\overline{t}_{uv}^{(r)} = t_{uw_0^{(r)}} + \sum_{l=1}^{L-1} t_{w_{l-1}^{(r)}w_l^{(r)}} + t_{w_{L-1}^{(r)}v}. \tag{8}$$

The variance of travelling time of the object moving from u to v along path r is

$$\overline{s}_{uv}^{(r)} = s_{uw_0^{(r)}} + \sum_{l=1}^{L-1} s_{w_{l-1}^{(r)}w_l^{(r)}} + s_{w_{L-1}^{(r)}v}. \tag{9}$$

Equations (6), (8), and (9) define a composite parameter matrix $\overline{A}$ with the same size as A. Each entry of $\overline{A}$ has

R_{uv}^{L} elements, and the rth element is composed of $\overline{\pi}_{uv}^{(r)}$, $\overline{t}_{uv}^{(r)}$, and $\overline{s}_{uv}^{(r)}$. If the Gaussian assumption does not hold, the composite parameter matrix $\overline{A}$ cannot be constituted directly from A. In this case, $\overline{A}$ should be established manually. For example, if we assume that the traveling time between two directly connected cameras follows the log-normal distribution, which is useful for modeling the object's long stay between cameras, the total traveling time along a specific path has no closed-form expression, but can be reasonably approximated by another log-normal distribution. A commonly used approximation is obtained by matching the mean and variance [23].

The model defined by (5)–(7) can be considered as a high-order probabilistic model in that it is capable of describing object's transitions between nonadjacent nodes in the camera networks. The order of the model is determined by the path length L.

3.4. Graphical Representation. Dynamic Bayesian networks model probabilistic time series as a directed graph, where the nodes represent random variables and directed edges correspond to conditional probability distributions. Figure 2 shows the dynamic Bayesian networks model of the data association problem in VSN.

In Figure 2 the arrows directed to z_i are defined by (2); the arrows directed to y_i are defined by (3) and (7). To complete the model, we set $z_1^{(k)} = 0$, for $k = 1, \ldots, K$.

4. Exact Inference

Based on the dynamic Bayesian networks model shown in Figure 2, data association problem in VSN can be solved by inferring the posterior marginal distribution of labeling variable $p(x_i \mid y_{0:i})$ from the observations and selecting the label with the highest posterior probability. In this section we present the exact inference algorithms, including forward pass and backward pass, then show the intractability of the exact inference when the number of objects is large.

4.1. Forward Pass for Exact Inference. From Figure 2 we can see that ω_i plays a role within a single time slice in DBN model, thus we define the belief state as the joint posterior probability of x_i and z_i and update it recursively based on the observation y_i at each time instance. Having the state transition model and observation model in hand, this is a standard state estimation problem. From Bayesian rule, the forward pass belief state can be written as

$$\begin{aligned} p(x_i, z_i \mid y_{0:i}) &= \sum_{\omega_i} p(x_i, z_i, \omega_i \mid y_{0:i}) \\ &= \frac{1}{L_i} \sum_{\omega_i} p(y_i \mid x_i, z_i, \omega_i, y_{0:i-1})\, p(x_i, z_i, \omega_i \mid y_{0:i-1}) \\ &= \frac{1}{L_i} \lambda_i(x_i = k)\, \eta_i\left(z_i^{(k)} = l\right) p(z_i \mid y_{0:i-1}), \end{aligned} \tag{10}$$

where $L_i = p(y_i \mid y_{0:i-1})$ is the normalizing constant. The appearance and spatiotemporal information are injected into the model through the terms λ_i and η_i, respectively, which are defined as follows:

$$\begin{aligned} \lambda_i(x_i = k) &= p(x_i = k)\, p(o_i \mid x_i = k) \quad \text{for } k = 1, \ldots, K, \\ \eta_i\left(z_i^{(k)} = l\right) &= \sum_{\omega_i^{(u,v)}} p\left(\omega_i^{(u,v)}\right) \\ &\quad \times p\left(d_i \mid x_i = k, z_i^{(k)} = l, \omega_i^{(u,v)}, \right. \\ &\quad \left. d_l, c_l = u, c_i = v\right) \quad \text{for } l = 0, \ldots, i-1. \end{aligned} \tag{11}$$

Note that the probability items corresponding to all elements in ω_i except $\omega_i^{(u,v)}$ are summed to one and $\omega_i^{(u,v)}$ is completely encapsulated in the term η_i. It turns out at this point that introducing ω_i results in a mixed spatiotemporal observation model, as it can be expressed in terms of a weighted sum of probabilities conditioned on different paths. To calculate the predictive probability of z_i, we first calculate the predictive probability of the joint state (z_i, x_{i-1}) and then marginalize x_{i-1} out. It can be written as

$$\begin{aligned} &p(z_i, x_{i-1} = j \mid y_{0:i-1}) \\ &\quad = \sum_{z_{i-1}} f(z_i \mid x_{i-1}, z_{i-1})\, p(x_{i-1} = j, z_{i-1} \mid y_{0:i-1}) \\ &\quad = \begin{cases} \displaystyle\sum_{m=0}^{i-2} p\left(x_{i-1} = j, z_{i-1}^{(j)} = m, z_{i-1}^{(\neg j)} = z_i^{(\neg j)} \mid y_{0:i-1}\right) \\ \quad \text{if } z_i^{(j)} = i-1, z_i^{(\neg j)} = 0, \ldots, i-2, \\ \text{otherwise} \\ \quad 0, \end{cases} \end{aligned} \tag{12}$$

where $z_i^{(\neg j)} \triangleq z_i^{(1:K)} \setminus z_i^{(j)}$. From the deterministic relationship of (2), if $x_{i-1} = j$, the summands in the first line of (12) are nonzero only when $z_{i-1}^{(\neg j)} = z_i^{(\neg j)}$. The last line of (12) ensures that all $z_i^{(k)}$ cannot be less than $i-1$ simultaneously and only $z_i^{(j)}$ can be equal to $i-1$ if $x_{i-1} = j$.

4.2. Backward Pass for Exact Inference. In backward inference, future observations can be used to further modify the estimation of current state. Following the similar manner of derivation in [24], and exploiting the conditional independency encoded in the DBN model shown in Figure 2, the backward belief state can be written as

$$
\begin{aligned}
p(x_i, z_i \mid y_{0:N}) &= \sum_{x_{i+1}} \sum_{z_{i+1}} p(x_i, z_i, x_{i+1}, z_{i+1} \mid y_{0:N}) \\
&= \sum_{x_{i+1}} \sum_{z_{i+1}} \frac{p(x_i, z_i \mid x_{i+1}, z_{i+1}, y_{0:i+1})\, p(x_{i+1}, z_{i+1} \mid y_{0:N})\, p(y_{0:N})}{p(y_{0:N})} \\
&= \sum_{x_{i+1}} \sum_{z_{i+1}} \frac{p(y_{i+1} \mid x_{i+1}, z_{i+1}, y_{0:i})\, p(x_{i+1}, z_{i+1} \mid x_i, z_i)\, p(x_i, z_i \mid y_{0:i})\, p(y_{0:i})}{p(x_{i+1}, z_{i+1} \mid y_{0:i+1})\, p(y_{0:i+1})}\, p(x_{i+1}, z_{i+1} \mid y_{0:N}) \\
&= \frac{1}{L_{i+1}}\, p(x_i, z_i \mid y_{0:i}) \sum_{x_{i+1}} \sum_{z_{i+1}} \lambda_{i+1}(x_{i+1} = k)\, \eta_{i+1}\left(z_{i+1}^{(k)} = l\right) f(z_{i+1} \mid x_i, z_i)\, \frac{p(x_{i+1}, z_{i+1} \mid y_{0:N})}{p(x_{i+1}, z_{i+1} \mid y_{0:i+1})} \\
&= \frac{1}{L_{i+1}}\, p(x_i, z_i \mid y_{0:i}) \sum_{x_{i+1}} \lambda_{i+1}(x_{i+1} = k)\, \eta_{i+1}\left(\bar{z}_{i+1}^{(k)}\left(x_i, z_i^{(k)}\right)\right) \frac{p(x_{i+1}, \bar{z}_{i+1}(x_i, z_i) \mid y_{0:N})}{p(x_{i+1}, \bar{z}_{i+1}(x_i, z_i) \mid y_{0:i+1})}.
\end{aligned}
\tag{13}
$$

Note that the normalizing constant in (13) is already available and $\bar{z}_{i+1}^{(k)}$ is a function of x_i and $z_i^{(k)}$, which is defined by (2).

Although the deterministic relation in (2) has simplified the inference computation significantly, it is clear in (10) and (13) that maintaining both forward and backward belief state is still intractable as the joint state space is the Cartesian product of the state space of x_i and K spaces of all $z_i^{(k)}$. At step i of forward passing, for example, there are Ki^K elements which need to be evaluated for updating the belief state. To make the inference practicable, we have to resort to approximate inference.

5. Approximate Inference

The basic idea of approximate inference is factorization. By factorizing the joint belief state into the product of several distributions of smaller sets of random variables, the memory and computational resources required for storing and updating belief state can be reduced. Inevitably, this factorization will introduce errors in belief state representation if the random variables in different sets are not indeed independent. However, Boyen and Koller [17] showed that, in terms of the Kullback-Leibler divergence, the inference error introduced by factorized representation of the belief state of discrete stochastic process is not accumulated infinitely over time. Furthermore, if the factorization is tailored to the specific structure of the process, the error has a bound determined by the minimum mixing rate of the involved subprocesses and the interaction among them. Theoretical results in [25] showed that using conditional independent clusters for approximate representation yields tighter bound. Although the theoretical results have not been extended to general stochastic process including continuous variables and to the case of reasoning backward in time, it is clearly suggested that for approximate inference, the structure of DBN may be exploited for computational gain in these circumstances. Following this line, in this section we present two kinds of factorization schemes based on the structure of DBN shown in Figure 2 and provide the corresponding forward and backward algorithms. The effect of the algorithms is shown in Section 7 with simulations and experiments.

The intractability of exact inference of our problem comes from the interdependency between variables. "Active path" [26] is a convenient tool for analyzing the dependence structure in belief networks: an active path from node i to j given node set K is a simple trail between i and j, such that every node with two incoming arcs on the trail is or has a descendant in K and every other node on the trail is not functionally determined by K. Two nodes are interdependent if they are connected by an active path. In Figure 2 we can identify the following two kinds of active paths: (a) active paths within a single time slice, $z_i^{(j)}$ and $z_i^{(k)}$ are coupled through the path $z_i^{(j)} - y_i - z_i^{(k)}$, and x_i and $z_i^{(k)}$ are coupled through $x_i - y_i - z_i^{(k)}$; (b) active paths across the past time slices, and $z_i^{(j)}$ and $z_i^{(k)}$ are coupled through the paths $z_i^{(j)} - x_{i-1} - z_i^{(k)}$ and $z_i^{(j)} - z_{i-1}^{(j)} - y_{i-1} - z_{i-1}^{(k)} - z_i^{(k)}$, and the longer paths $z_i^{(j)} - z_{i-1}^{(j)} - x_{i-2} - z_{i-1}^{(k)} - z_i^{(k)}$ and $z_i^{(j)} - z_{i-1}^{(j)} - z_{i-2}^{(j)} - y_{i-2} - z_{i-2}^{(k)} - z_{i-1}^{(k)} - z_i^{(k)}$, and so on. It should be noticed, however, that the active paths between $z_i^{(k)}$s can be disconnected if the value of x at proper time slice is given. For example, $z_i^{(j)} - y_i - z_i^{(k)}$ breaks if x_i is given; the pair of paths $z_i^{(j)} - x_{i-1} - z_i^{(k)}$ and $z_i^{(j)} - z_{i-1}^{(j)} - y_{i-1} - z_{i-1}^{(k)} - z_i^{(k)}$ break if x_{i-1} is given, and so on.

In Section 5.1, we present a simple approximate inference approach based on the marginal independence assumption which naively neglects all the active paths mentioned above. In Section 5.2 we propose another approximate inference which neglects the active paths across the past time slices and preserves the path within the current time slice and factorizes the joint belief state based on the assumed conditional independence. In simulations the second approximate inference demonstrates better compromise between inference accuracy and computational efficiency. In Section 5.3 we discuss the problem of choice of active path for approximate inference in more detail and the relationship with other works.

5.1. Approximate Inference I. In the first factorization scheme, the joint belief state is naively decomposed into the product of marginal distributions of x_i and $z_i^{(k)}$, that is,

$$\begin{aligned} p(x_i, z_i \mid y_{0:i}) &\approx \widetilde{p}(x_i, z_i \mid y_{0:i}) \\ &= \widehat{p}(x_i \mid y_{0:i}) \prod_{k=1}^{K} \widehat{p}\left(z_i^{(k)} \mid y_{0:i}\right), \\ p(x_i, z_i \mid y_{0:N}) &\approx \widetilde{p}(x_i, z_i \mid y_{0:N}) \\ &= \widehat{p}(x_i \mid y_{0:N}) \prod_{k=1}^{K} \widehat{p}\left(z_i^{(k)} \mid y_{0:N}\right). \end{aligned} \tag{14}$$

At step i in the forward pass, the approximate belief state $\widetilde{p}(x_{i-1}, z_{i-1} \mid y_{0:i-1})$ is propagated through the transition model, obtaining $\widehat{p}(x_i, z_i \mid y_{0:i-1})$, and conditioned on the current observation, obtaining $\widehat{p}(x_i, z_i \mid y_{0:i})$, then approximated by (14), obtaining $\widetilde{p}(x_i, z_i \mid y_{0:i})$. The process of backward pass is similar.

5.1.1. Forward Pass in Approximate Inference I. To derive the forward pass algorithm, we first calculate the marginal distributions $\widehat{p}_i$ in (14) from (10) and then try to express them in terms of the marginal distributions of the last time instance $\widehat{p}_{i-1}$ based on the independence assumption. The marginal distribution of x_i is

$$\begin{aligned} &\widehat{p}(x_i = k \mid y_{0:i}) \\ &\quad = \sum_{z_i} \widehat{p}(x_i = k, z_i \mid y_{0:i}) \\ &\quad = \frac{1}{L_i^f} \lambda_i(x_i = k) \sum_{z_i} \eta_i\left(z_i^{(k)} = l\right) \widehat{p}(z_i \mid y_{0:i-1}) \\ &\quad = \frac{1}{L_i^f} \lambda_i(x_i = k) \sum_{z_i^{(k)}} \eta_i\left(z_i^{(k)} = l\right) \widehat{p}\left(z_i^{(k)} = l \mid y_{0:i-1}\right). \end{aligned} \tag{15}$$

For $k = 1, \ldots, K$, the marginal distribution of $z_i^{(k)}$ is

$$\begin{aligned} &\widehat{p}\left(z_i^{(k)} = l \mid y_{0:i}\right) \\ &\quad = \sum_{x_i} \sum_{z_i^{(\neg k)}} \widehat{p}(x_i, z_i \mid y_{0:i}) \\ &\quad = \frac{1}{L_i^f} \sum_{x_i} \sum_{z_i^{(\neg k)}} \lambda_i(x_i = k) \eta_i\left(z_i^{(k)} = l\right) \widehat{p}(z_i \mid y_{0:i-1}) \\ &\quad = \frac{1}{L_i^f} \lambda_i(x_i = k) \eta_i\left(z_i^{(k)} = l\right) \widehat{p}\left(z_i^{(k)} = l \mid y_{0:i-1}\right) \\ &\qquad + \frac{1}{L_i^f} \sum_{\substack{x_i = 1 \\ x_i \neq k}}^{K} \lambda_i(x_i = j) \sum_{z_i^{(j)}} \eta_i\left(z_i^{(j)} = m\right) \\ &\qquad\qquad \times \widehat{p}\left(z_i^{(j)} = m, z_i^{(k)} = l \mid y_{0:i-1}\right). \end{aligned} \tag{16}$$

There are two kinds of predictive distributions in (15) and (16), one is over single $z_i^{(k)}$, and the other is over the pair $(z_i^{(j)}, z_i^{(k)})$. We first calculate the joint predictive probabilities of them with x_{i-1}, then marginalize x_{i-1} out. The joint predictive distribution of $(z_i^{(k)}, x_{i-1})$ is

$$\begin{aligned} &\widehat{p}\left(z_i^{(k)} = l, x_{i-1} = n \mid y_{0:i-1}\right) \\ &\quad = \sum_{z_{i-1}^{(k)}} f\left(z_i^{(k)} = l \mid x_{i-1} = n, z_{i-1}^{(k)}\right) \widetilde{p}\left(x_{i-1} = n, z_{i-1}^{(k)} \mid y_{0:i-1}\right) \\ &\quad = \begin{cases} \widehat{p}(x_{i-1} = k \mid y_{0:i-1}) & \text{if } n = k,\ l = i-1, \\ \widehat{p}(x_{i-1} = n \mid y_{0:i-1}) & \\ \quad \times \widehat{p}\left(z_{i-1}^{(k)} = l \mid y_{0:i-1}\right) & \text{if } n \neq k,\ l = 0 : i-2, \\ \text{otherwise} & 0. \end{cases} \end{aligned} \tag{17}$$

The joint predictive distribution of $(z_i^{(k)}, z_i^{(j)}, x_{i-1})$ is

$$\begin{aligned} &\widehat{p}\left(z_i^{(j)} = m, z_i^{(k)} = l, x_{i-1} = n \mid y_{0:i-1}\right) \\ &\quad = \sum_{z_{i-1}^{(j)}} \sum_{z_{i-1}^{(k)}} f\left(z_i^{(j)} = m, z_i^{(k)} = l \mid x_{i-1} = n, z_{i-1}^{(j)}, z_{i-1}^{(k)}\right) \\ &\qquad \times \widetilde{p}\left(x_{i-1} = n, z_{i-1}^{(j)}, z_{i-1}^{(k)} \mid y_{0:i-1}\right) \\ &\quad = \begin{cases} \widehat{p}(x_{i-1} = j \mid y_{0:i-1}) \widehat{p}\left(z_{i-1}^{(k)} = l \mid y_{0:i-1}\right) & \\ \quad \text{if } n = j,\ m = i-1,\ l = 0 : i-2, & \\ \widehat{p}(x_{i-1} = k \mid y_{0:i-1}) \widehat{p}\left(z_{i-1}^{(j)} = m \mid y_{0:i-1}\right) & \\ \quad \text{if } n = k,\ m = 0 : i-2,\ l = i-1, & \\ \widehat{p}(x_{i-1} = n \mid y_{0:i-1}) \widehat{p}\left(z_{i-1}^{(j)} = m \mid y_{0:i-1}\right) & \\ \quad \times \widehat{p}\left(z_{i-1}^{(k)} = l \mid y_{0:i-1}\right) & \\ \quad \text{if } n \neq j,\ n \neq k,\ m = 0 : i-2,\ l = 0 : i-2, & \\ \text{otherwise} & \\ \quad 0. & \end{cases} \end{aligned} \tag{18}$$

Note that the independence assumption in (14) plays its role in the last line of (17) and (18). To update the belief state at step i using (15)–(18), we only need to evaluate the probabilities of $K + Ki$ different configurations. The computation is greatly simplified. The forward pass algorithm for *approximate inference I* is depicted graphically in Figure 3.

5.1.2. Backward Pass in Approximate Inference I. The derivation of backward pass algorithm is straightforward. We first substitute (14) into (13), obtaining

$$\begin{aligned}
&\widehat{p}(x_i, z_i \mid y_{0:N}) \\
&\quad = \frac{1}{L_{i+1}^{b}} \widetilde{p}(x_i, z_i \mid y_{0:i}) \\
&\qquad \times \sum_{x_{i+1}} \lambda_{i+1}(x_{i+1} = k) \eta_{i+1}\left(\overline{z}_{i+1}^{(k)}\left(x_i, z_i^{(k)}\right)\right) \\
&\qquad \times \frac{\widetilde{p}(x_{i+1}, \overline{z}_{i+1}(x_i, z_i) \mid y_{0:N})}{\widetilde{p}(x_{i+1}, \overline{z}_{i+1}(x_i, z_i) \mid y_{0:i+1})} \\
&\quad = \frac{1}{L_{i+1}^{b}} \widehat{p}(x_i \mid y_{0:i}) \prod_{\tau} \widehat{p}\left(z_i^{(\tau)} \mid y_{0:i}\right) \\
&\qquad \cdot \sum_{x_{i+1}} \lambda_{i+1}(x_{i+1} = k) \eta_{i+1}\left(\overline{z}_{i+1}^{(k)}\left(x_i, z_i^{(k)}\right)\right) \\
&\qquad \times \frac{\widehat{p}(x_{i+1} \mid y_{0:N})}{\widehat{p}(x_{i+1} \mid y_{0:i+1})} \prod_{\tau} \frac{\widehat{p}\left(\overline{z}_{i+1}^{(\tau)}\left(x_i, z_i^{(\tau)}\right) \mid y_{0:N}\right)}{\widehat{p}\left(\overline{z}_{i+1}^{(\tau)}\left(x_i, z_i^{(\tau)}\right) \mid y_{0:i+1}\right)}.
\end{aligned} \tag{19}$$

Note that in approximate inference the normalization constant $L_{i+1}^{b} \neq L_{i+1}^{f}$. Then we calculate the marginal distribution of x_i

$$\begin{aligned}
&\widehat{p}(x_i = n \mid y_{0:N}) \\
&\quad = \sum_{z_i} \widehat{p}(x_i, z_i \mid y_{0:N}) \\
&\quad = \frac{1}{L_{i+1}^{b}} \widehat{p}(x_i = n \mid y_{0:i}) \\
&\qquad \times \sum_{x_{i+1}} \overline{\lambda}_{i+1}(x_{i+1} = k) \prod_{\tau} \sum_{z_i^{(\tau)}} \overline{\eta}_{i+1}^{(k)}\left(x_i, z_i^{(\tau)}\right) \phi_{i+1}\left(x_i, z_i^{(\tau)}\right)
\end{aligned} \tag{20}$$

and the marginal distribution of $z_i^{(k)}$

$$\begin{aligned}
&\widehat{p}\left(z_i^{(j)} = m \mid y_{0:N}\right) \\
&\quad = \sum_{x_i} \sum_{z_i^{(\neg j)}} \widehat{p}(x_i, z_i \mid y_{0:N}) \\
&\quad = \frac{1}{L_{i+1}^{b}} \sum_{x_{i+1}} \overline{\lambda}_{i+1}(x_{i+1} = k) \\
&\qquad \cdot \sum_{x_i} \widehat{p}(x_i \mid y_{0:i}) \overline{\eta}_{i+1}^{(k)}\left(x_i, z_i^{(j)} = m\right) \phi_{i+1}\left(x_i, z_i^{(j)} = m\right) \\
&\qquad\quad \times \prod_{\tau \neq j} \sum_{z_i^{(\tau)}} \overline{\eta}_{i+1}^{(k)}\left(x_i, z_i^{(\tau)}\right) \phi_{i+1}\left(x_i, z_i^{(\tau)}\right),
\end{aligned} \tag{21}$$

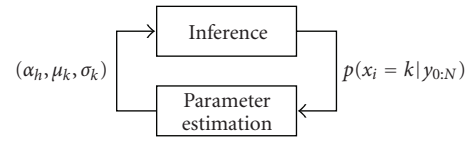

Figure 5: The EM framework. The inference module is implemented with the algorithms presented in Sections 4 and 5, and the parameter estimation module is implemented with (34)–(36).

where the terms $\overline{\lambda}_{i+1}$, $\overline{\eta}_{i+1}^{(k)}$, and ϕ_{i+1} are defined as

$$\overline{\lambda}_{i+1}(x_{i+1} = k) = \lambda_{i+1}(x_{i+1} = k) \frac{\widehat{p}(x_{i+1} \mid y_{0:N})}{\widehat{p}(x_{i+1} \mid y_{0:i+1})}, \tag{22}$$

$$\begin{aligned}
&\overline{\eta}_{i+1}^{(k)}\left(x_i = n, z_i^{(\tau)} = l\right) \\
&\quad = \begin{cases} 1 & \text{if } \tau \neq k, \\ \eta_{i+1}\left(z_{i+1}^{(k)} = i\right) & \text{if } \tau = k,\ n = k, \\ \eta_{i+1}\left(z_{i+1}^{(k)} = l\right) & \text{if } \tau = k,\ n \neq k, \end{cases}
\end{aligned} \tag{23}$$

$$\begin{aligned}
&\phi_{i+1}\left(x_i = n, z_i^{(\tau)} = l\right) \\
&\quad = \begin{cases} \dfrac{\widehat{p}\left(z_{i+1}^{(\tau)} = i \mid y_{0:N}\right)}{\widehat{p}\left(z_{i+1}^{(\tau)} = i \mid y_{0:i+1}\right)} \widehat{p}\left(z_i^{(\tau)} = l \mid y_{0:i}\right) & \text{if } n = \tau, \\ \dfrac{\widehat{p}\left(z_{i+1}^{(\tau)} = l \mid y_{0:N}\right)}{\widehat{p}\left(z_{i+1}^{(\tau)} = l \mid y_{0:i+1}\right)} \widehat{p}\left(z_i^{(\tau)} = l \mid y_{0:i}\right) & \text{if } n \neq \tau. \end{cases}
\end{aligned} \tag{24}$$

5.2. Approximate Inference II. In the second factorization scheme, we preserve the interdependence between x_i and z_i and assume that $z_i^{(j)}$ and $z_i^{(k)}$ are conditional independent given x_i. Then the joint belief state is decomposed as

$$\begin{aligned}
p(x_i, z_i \mid y_{0:i}) &\approx \widetilde{p'}(x_i, z_i \mid y_{0:i}) \\
&= \widehat{p'}(x_i \mid y_{0:i}) \prod_{k=1}^{K} \widehat{p'}\left(z_i^{(k)} \mid x_i, y_{0:i}\right), \\
p(x_i, z_i \mid y_{0:N}) &\approx \widetilde{p'}(x_i, z_i \mid y_{0:N}) \\
&= \widehat{p'}(x_i \mid y_{0:N}) \prod_{k=1}^{K} \widehat{p'}\left(z_i^{(k)} \mid x_i, y_{0:N}\right).
\end{aligned} \tag{25}$$

The process of forward and backward pass is the same as before, except for the approximation manner of the belief state.

5.2.1. Forward Pass in Approximate Inference II. To write down the forward pass algorithm for belief state representation in (25), we need to compute the marginal distributions $\widehat{p'}$ over x_i and $(x_i, z_i^{(k)})$. The former can be calculated as in

(15), but with different definition of $\widehat{p'}(z_i^{(k)} \mid y_{0:i-1})$. The latter can be written as

$$\begin{aligned}
&\widehat{p'}\left(x_i = j, z_i^{(k)} = l \mid y_{0:i}\right) \\
&\quad = \sum_{z_i^{(\neg k)}} \widehat{p'}\left(x_i = j, z_i^{(k)} = l, z_i^{(\neg k)} \mid y_{0:i}\right) \\
&\quad = \frac{1}{L_i^f}\lambda_i(x_i = j) \sum_{z_i^{(\neg k)}} \eta_i\left(z_i^{(j)} = m\right)\widehat{p'}(z_i \mid y_{0:i-1}) \\
&\quad = \begin{cases} \dfrac{1}{L_i^f}\lambda_i(x_i = k)\eta_i\left(z_i^{(k)} = l\right) & \\ \quad \times \widehat{p'}\left(z_i^{(k)} = l \mid y_{0:i-1}\right) & \text{if } j = k, \\ \dfrac{1}{L_i^f}\lambda_i(x_i = j)\sum\limits_{z_i^{(j)}} \eta_i\left(z_i^{(j)} = m\right) & \\ \quad \times \widehat{p'}\left(z_i^{(j)} = m, z_i^{(k)} = l \mid y_{0:i-1}\right) & \text{if } j \neq k. \end{cases}
\end{aligned} \tag{26}$$

Based on the independence assumption in (25), the two predictive distributions in (17) and (18) are redefined as

$$\begin{aligned}
&\widehat{p'}\left(z_i^{(k)} = l, x_{i-1} = n \mid y_{0:i-1}\right) \\
&= \sum_{z_{i-1}^{(k)}} f\left(z_i^{(k)} = l \mid x_{i-1} = n, z_{i-1}^{(k)}\right)\widetilde{p'}\left(x_{i-1} = n, z_{i-1}^{(k)} \mid y_{0:i-1}\right) \\
&= \begin{cases} \widehat{p'}(x_{i-1} = k \mid y_{0:i-1}) & \text{if } n = k,\ l = i-1, \\ \widehat{p'}\left(x_{i-1} = n, z_{i-1}^{(k)} = l \mid y_{0:i-1}\right) & \text{if } n \neq k,\ l = 0 : i-2, \\ \text{otherwise} & 0, \end{cases}
\end{aligned} \tag{27}$$

$$\begin{aligned}
&\widehat{p'}\left(z_i^{(j)} = m, z_i^{(k)} = l, x_{i-1} = n \mid y_{0:i-1}\right) \\
&= \sum_{z_{i-1}^{(j)}}\sum_{z_{i-1}^{(k)}} f\left(z_i^{(j)} = m, z_i^{(k)} = l \mid x_{i-1} = n, z_{i-1}^{(j)}, z_{i-1}^{(k)}\right) \\
&\qquad \times \widetilde{p'}\left(x_{i-1} = n, z_{i-1}^{(j)}, z_{i-1}^{(k)} \mid y_{0:i-1}\right) \\
&= \begin{cases} \widehat{p'}\left(x_{i-1} = j, z_{i-1}^{(k)} = l \mid y_{0:i-1}\right) & \\ \quad \text{if } n = j,\ m = i-1,\ l = 0 : i-2, & \\ \widehat{p'}\left(x_{i-1} = k, z_{i-1}^{(j)} = m \mid y_{0:i-1}\right) & \\ \quad \text{if } n = k,\ m = 0 : i-2,\ l = i-1, & \\ \dfrac{\widehat{p'}\left(z_{i-1}^{(j)} = m, x_{i-1} = n \mid y_{0:i-1}\right)\widehat{p'}}{\widehat{p'}(x_{i-1} = n \mid y_{0:i-1})} & \\ \quad \times\left(z_{i-1}^{(k)} = l, x_{i-1} = n \mid y_{0:i-1}\right) & \\ \quad \text{if } n \neq j,\ n \neq k,\ m = 0 : i-2,\ l = 0 : i-2, & \\ \text{otherwise} & \\ \quad 0. & \end{cases}
\end{aligned} \tag{28}$$

When belief state is updated by (26)–(28) at step i, $K + K^2 i$ elements need to be evaluated. The forward pass algorithm for *approximate inference II* is depicted graphically in Figure 4. Although the computational burden increases to some extent compared with *approximation inference I* (but still much less than that in exact algorithm), simulation results show that the inference accuracy is improved significantly, approaching that of the exact algorithm.

5.2.2. Backward Pass in Approximate Inference II. As before, to derive the backward pass algorithm for *approximate inference II*, we substitute (25) into (13), obtaining

$$\begin{aligned}
&\widehat{p'}(x_i, z_i \mid y_{0:N}) \\
&\quad = \frac{1}{L_{i+1}^b}\widehat{p'}(x_i \mid y_{0:i})\prod_{\tau}\widehat{p'}\left(z_i^{(\tau)} \mid x_i, y_{0:i}\right) \\
&\qquad \cdot \sum_{x_{i+1}} \lambda_{i+1}(x_{i+1} = k)\eta_{i+1}\left(\overline{z}_{i+1}^{(k)}\left(x_i, z_i^{(k)}\right)\right)\frac{\widehat{p'}(x_{i+1} \mid y_{0:N})}{\widehat{p'}(x_{i+1} \mid y_{0:i+1})} \\
&\qquad \times \prod_{\tau}\frac{\widehat{p'}\left(\overline{z}_{i+1}^{(\tau)}\left(x_i, z_i^{(\tau)}\right) \mid x_{i+1}, y_{0:N}\right)}{\widehat{p'}\left(\overline{z}_{i+1}^{(\tau)}\left(x_i, z_i^{(\tau)}\right) \mid x_{i+1}, y_{0:i+1}\right)},
\end{aligned} \tag{29}$$

then calculate the marginal distribution of x_i

$$\begin{aligned}
&\widehat{p'}(x_i = n \mid y_{0:N}) \\
&\quad = \sum_{z_i}\widehat{p'}(x_i, z_i \mid y_{0:N}) \\
&\quad = \frac{1}{L_{i+1}^b}\widehat{p'}(x_i = n \mid y_{0:i})\sum_{x_{i+1}}\overline{\lambda}_{i+1}(x_{i+1} = k) \\
&\qquad \times \prod_{\tau}\sum_{z_i^{(\tau)}}\overline{\eta}_{i+1}^{(k)}\left(x_i, z_i^{(\tau)}\right)\psi_{i+1}\left(x_i, z_i^{(\tau)}, x_{i+1}\right)
\end{aligned} \tag{30}$$

and the marginal distribution of $(x_i, z_i^{(k)})$

$$\begin{aligned}
&\widehat{p'}\left(x_i = n, z_i^{(j)} = m \mid y_{0:N}\right) \\
&\quad = \sum_{z_i^{(\neg j)}}\widehat{p'}(x_i, z_i \mid y_{0:N}) \\
&\quad = \frac{1}{L_{i+1}^b}\sum_{x_{i+1}}\overline{\lambda}_{i+1}(x_{i+1} = k)\overline{\eta}_{i+1}^{(k)}\left(x_i, z_i^{(j)}\right)\psi_{i+1}\left(x_i, z_i^{(j)}, x_{i+1}\right) \\
&\qquad \times \prod_{\tau \neq j_{z_i^{(\tau)}}}\sum\overline{\eta}_{i+1}^{(k)}\left(x_i, z_i^{(\tau)}\right)\psi_{i+1}\left(x_i, z_i^{(\tau)}, x_{i+1}\right),
\end{aligned} \tag{31}$$

where $\overline{\lambda}_{i+1}$, and $\overline{\eta}_{i+1}^{(k)}$ are defined by (22) and (23). ψ_{i+1} is defined as

$$\psi_{i+1}\left(x_i = n, z_i^{(\tau)} = l, x_{i+1} = k\right) = \begin{cases} \dfrac{\widehat{p'}\left(z_{i+1}^{(\tau)} = i, x_{i+1} = k \mid y_{0:N}\right)}{\widehat{p'}\left(z_{i+1}^{(\tau)} = i, x_{i+1} = k \mid y_{0:i+1}\right)} \cdot \dfrac{\widehat{p'}(x_{i+1} = k \mid y_{0:i+1})}{\widehat{p'}(x_{i+1} = k \mid y_{0:N})} \cdot \dfrac{\widehat{p'}\left(z_i^{(\tau)} = l, x_i = n \mid y_{0:i}\right)}{\widehat{p'}(x_i = n \mid y_{0:i})} & \text{if } n = \tau, \\ \dfrac{\widehat{p'}\left(z_{i+1}^{(\tau)} = l, x_{i+1} = k \mid y_{0:N}\right)}{\widehat{p'}\left(z_{i+1}^{(\tau)} = l, x_{i+1} = k \mid y_{0:i+1}\right)} \cdot \dfrac{\widehat{p'}(x_{i+1} = k \mid y_{0:i+1})}{\widehat{p'}(x_{i+1} = k \mid y_{0:N})} \cdot \dfrac{\widehat{p'}\left(z_i^{(\tau)} = l, x_i = n \mid y_{0:i}\right)}{\widehat{p'}(x_i = n \mid y_{0:i})}, & \text{if } n \neq \tau. \end{cases} \tag{32}$$

5.3. Discussion. In this section we will discuss the problem of choosing active path for approximate inference in more detail. Here we define the belief state as the joint distribution of $(z_i^{1:K}, x_{1:i})$ and the exact forward inference can be reformulated as

$$\begin{aligned} p\left(z_i^{1:K}, x_{1:i} \mid y_{0:i}\right) &= \frac{1}{L_i}\lambda_i(x_i = k)\eta_i\left(z_i^{(k)} = l\right) \\ &\quad \times p\left(z_i^{1:K} \mid x_{1:i-1}\right)p\left(x_{1:i-1} \mid y_{0:i-1}\right) \\ &= \frac{1}{L_i}\lambda_i(x_i = k)\eta_i\left(z_i^{(k)} = l\right) \\ &\quad \times \prod_{j=1}^{K} p\left(z_i^{(j)} \mid x_{1:i-1}\right)p\left(x_{1:i-1} \mid y_{0:i-1}\right), \end{aligned} \tag{33}$$

where λ_i and η_i are defined in (11). Note that the joint predictive distribution of $z_i^{1:K}$ is completely decomposable given $x_{1:i-1}$. In exact inference using (33), we have to enumerate in the sampling space of $x_{1:i-1}$. This just shifts the problem from the intractable enumeration of $z_i^{1:K}$ to that of $x_{1:i-1}$. For tractable inference, we must discard some of the conditioning variables $x_{1:i-1}$. Discarding all $x_{1:i-1}$ leads to the proposed approximate inference II.

Formulation (33) provides a clearer view of the "relative significance" of active path corresponding to variable x_τ, $\tau = 1 : i-1$. Note that with some probability δ_τ, $z_i^{(j)}$ is functionally determined by x_τ. In other words, the τth active path is disconnected by x_τ with probability δ_τ. Thus we can use δ_τ as a measure of the "relative significance" of the τth active paths. It is easy to show that δ_τ decreases exponentially as τ varying from $i-1$ to 1. In fact, for time slice i, the relative significance of $i-1$th path is $\delta_{i-1} = p(x_{i-1} = j)$, and the relative significance of $i-2$th path is $\delta_{i-2} = p(x_{i-1} \neq j)p(x_{i-2} = j)$, and so on (we omit the conditioning variables $y_{0:i-1}$ temporally for clarity). This fact implies that the "recent" active paths are far more important than the "ancient" ones for accurate inference as they are less likely to be disconnected. We delete the conditioning variables x_τ in (33) one by one from $\tau = 1$ to $i-1$, resulting in a set of approximate inference algorithms and compare them with the proposed approximate inference II. We observed in simulations that the conditioning variables x_τ earlier than $i-1$ have much less effect on inference accuracy than x_{i-1} and including x_{i-1} can only improve the inference accuracy to a limited extent, but at the cost of a significant increase in computational burden.

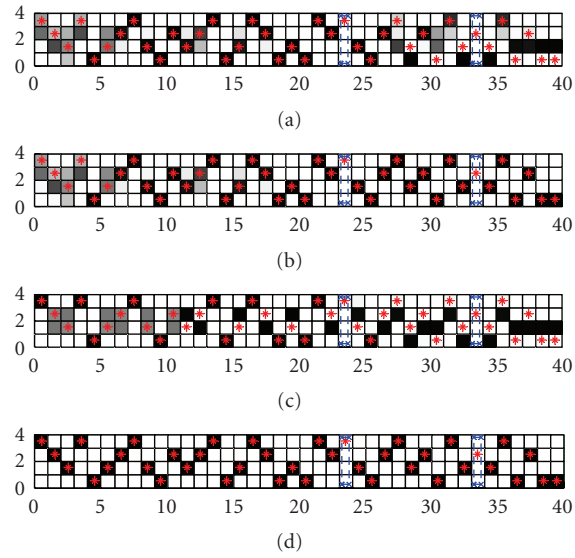

Figure 6: Marginal distribution of labeling variable in exact inference. The 24th and 34th observations are missed, depicted as dashed column. The true labels are depicted by stars. Each column represents the marginal distribution of the label of an observation. Grayscale corresponds to probability value. Black represents probability 1 and white probability 0. (a) Forward pass with 0-order model. (b) Forward pass with 1-order model. (c) Backward pass with 0-order model. (d) Backward pass with 1-order model.

It is interesting to relate our works with [27], where an approximate variational inference approach is proposed based on conditional entropy decomposition. As evaluating the negative entropy term in the objective function of the optimization problem is intractable if the graph size is large, and the authors decompose the full model into a sum of conditional entropies using the entropy chain rule, and then restrict the number of conditioning variables by discarding some of them. Since removing conditioning variables cannot decrease the entropy, this approximation leads to an upper bound of the objective function. In fact, in [27] the approximation of inference manifests in replacing the joint distribution of interest with a product of conditional distributions and discarding some of the conditioning variables based on the assumed conditional

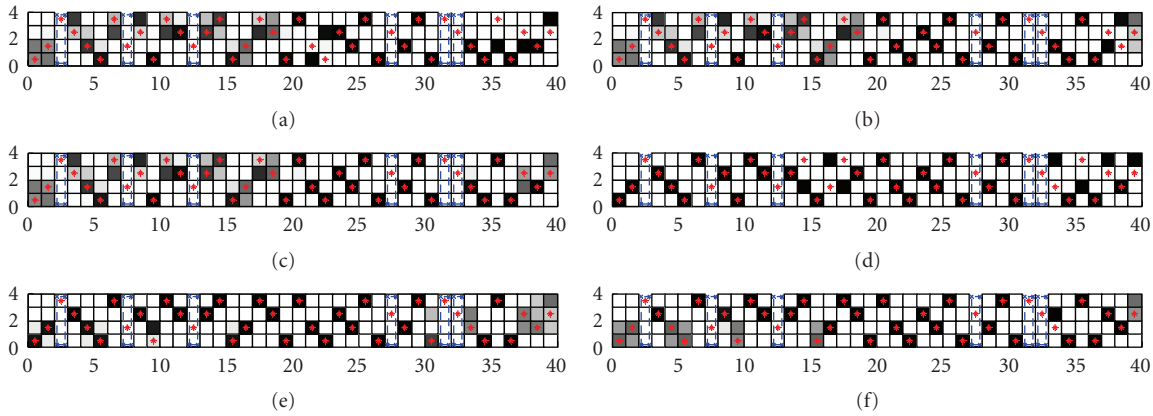

Figure 7: Marginal distribution of labeling variable in exact inference. The 3rd, 8th, 13th, 28th, 32nd, and 33rd observations are missed, depicted as dashed column. The true labels are depicted by stars. Each column represents the marginal distribution of the label of an observation. Grayscale corresponds to probability value. Blacks represent probability 1 and white probability 0. (a) Forward pass with 0-order model. (b) Forward pass with 1-order model. (c) Forward pass with 2-order model. (d) Backward pass with 0-order model. (e) Backward pass with 1-order model. (f) Backward pass with 2-order model.

independence, which is just the same scheme used in our approximate inference II. The authors in [27] point out that the more conditioning variables preserved, the tighter the bound is. This is also consistent with our results. However, it is not clear in [27] how to choose the conditioning variables in an optimal way. Besides, our approximate inference (I and II) is similar with the Factored Frontier algorithm [28] in that both of them factorize the joint belief state. But there is one important difference: our algorithms update the factored belief state from time $t-1$ to t exactly before computing the marginals in t, whereas Factored Frontier computes the approximate marginals directly based on additional independence assumption, resulting in more errors in calculation.

It is tempting in application that the independence structure can be discovered automatically. This enables the algorithm to choose the approximation scheme adaptively according to changing situations. We notice that in [29] an incremental thin junction tree algorithm is proposed which can adaptively choose a set of clusters and separators, that is, a junction tree, at each time step to approximate the belief state. We plan to incorporate this idea into our method in the future.

6. EM Framework for Unknown Appearance Parameters

In the previous discussion, we assumed that the appearance models of the objects under tracking are available. However, in typical scenarios of practical interests, the parameters of the appearance model are usually unknown and need to be estimated from observations. If we had known the label of each observation, that is, the object from which the observation is generated, the parameter estimation is straightforward. But the labels are also unknown and need to be estimated with data association algorithms. Considering the hidden labels as missing data, the problems of parameter estimation and data association can be solved simultaneously under the EM framework [30].

In this paper, the appearance observations are assumed to be generated from Gaussian mixture model, and the E-step and M-step in the EM framework take a very intuitive and simple form. We use $\Theta = \{\alpha_k, \mu_k, \sigma_k\}_{k=1}^{K}$ to denote the model parameters, where $\alpha_k \triangleq p(x_i = k)$ is the prior probability of the label k, μ_k and σ_k are mean and variance of appearance of object k. In E-step, based on the old guess of the model parameters Θ^{old}, we calculate the ownership probability of each observation, that is, the posterior probability of the hidden label corresponding to each observation, $p(x_i | y_{0:N}, \Theta^{\text{old}})$, with the data association algorithm presented in previous sections. Note that if we use forward passing algorithm for inference, the ownership probability is $p(x_i | y_{0:i}, \Theta^{\text{old}})$. In M-step, the model parameters are updated as in classical EM algorithm for Gaussian mixture model [31]

$$\begin{aligned}
\alpha_k^{\text{new}} &= \frac{1}{N}\sum_{i=1}^{N} p\left(x_i = k \mid y_{0:N}, \Theta^{\text{old}}\right),\\
\mu_k^{\text{new}} &= \frac{\sum_{i=1}^{N} o_i p\left(x_i = k \mid y_{0:N}, \Theta^{\text{old}}\right)}{\sum_{i=1}^{N} p\left(x_i = k \mid y_{0:N}, \Theta^{\text{old}}\right)},\\
\sigma_k^{\text{new}} &= \frac{\sum_{i=1}^{N} p\left(x_i = k \mid y_{0:N}, \Theta^{\text{old}}\right)\left(o_i - \mu_k^{\text{new}}\right)\left(o_i - \mu_k^{\text{new}}\right)'}{\sum_{i=1}^{N} p\left(x_i = k \mid y_{0:N}, \Theta^{\text{old}}\right)}.
\end{aligned} \tag{34}$$

The EM framework presented above is shown in Figure 5. It can be thought of as a generalization of the classical EM algorithm for the GMM [31] in that it calculates the ownership probabilities with inference algorithms based on both appearance and spatiotemporal information, instead of calculating them with Bayes rule based solely on the current appearance observation. Note that the EM proposed above can only work in an offline manner. Yet the learnt

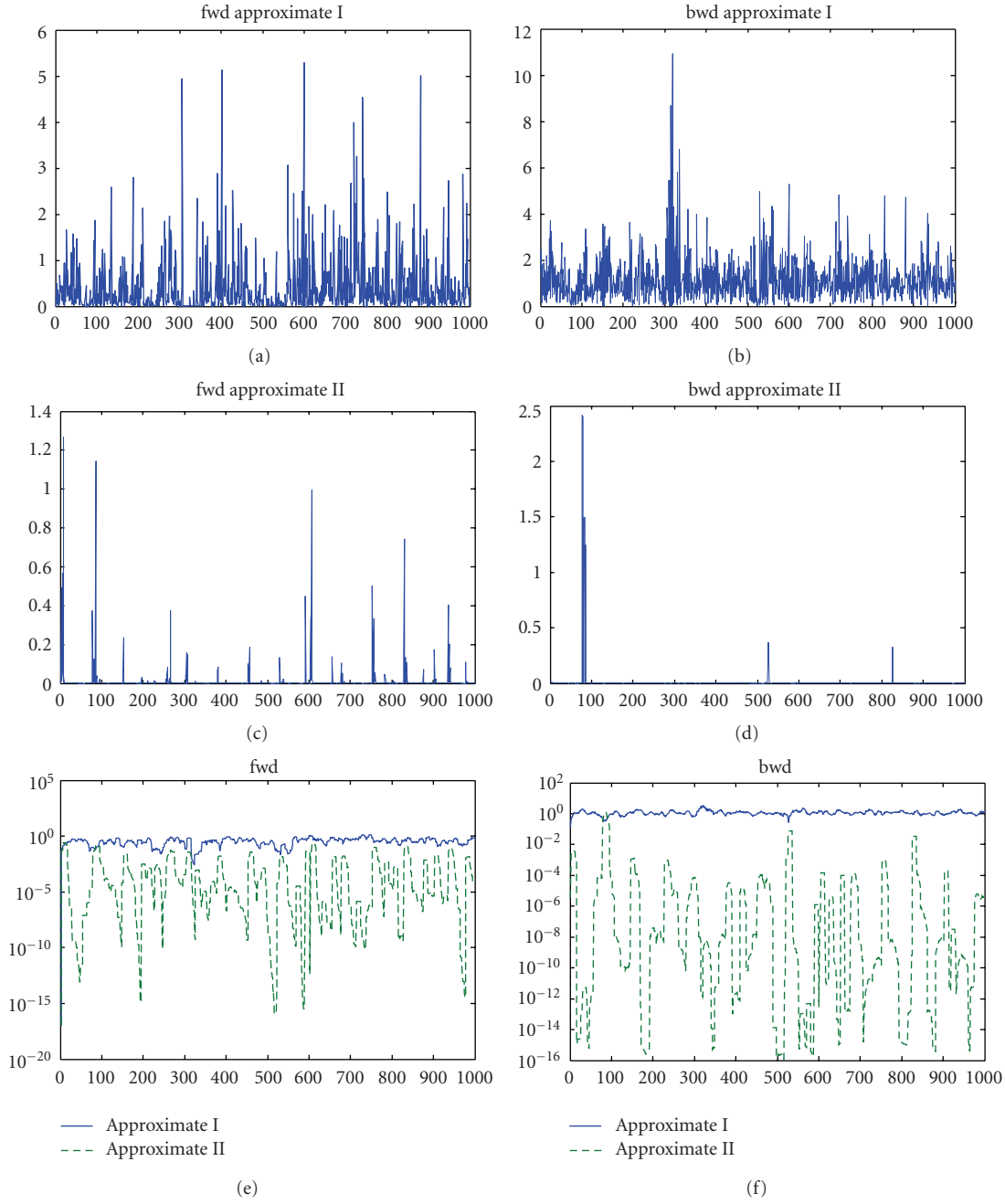

Figure 8: KL divergence caused by approximate inference. (a) Forward approximate I. (b) Backward approximate I. (c) Forward approximate II. (d) Backward approximate II. (e) Comparison between smoothed KL divergence of forward approximates I and II in log domain. (f) Comparison between smoothed KL divergence of backward approximates I and II in log domain.

model can be used for online data association using forward inference algorithms. In the future, we plan to investigate how to incorporate the inference engine into online EM such that both model learning and data association can be accomplished simultaneously on the fly. It is interested if we could estimate the parameters in the spatiotemporal model using EM as well. However, in M step, it is difficult to find an algorithm similar to (34) to update the guessed value of spatiotemporal parameters due to the existence of missing detections.

7. Results

7.1. Simulations

7.1.1. Data. To generate simulation data, we use the VSN with topological model shown as Figure 1 and specify the parameter matrix A of VSN and the appearance model parameters (μ_k, σ_k^2) of each object under tracking. The mean travel time t_{uv} between adjacent nodes u and v is proportional to their distance, as shown in Table 1. And

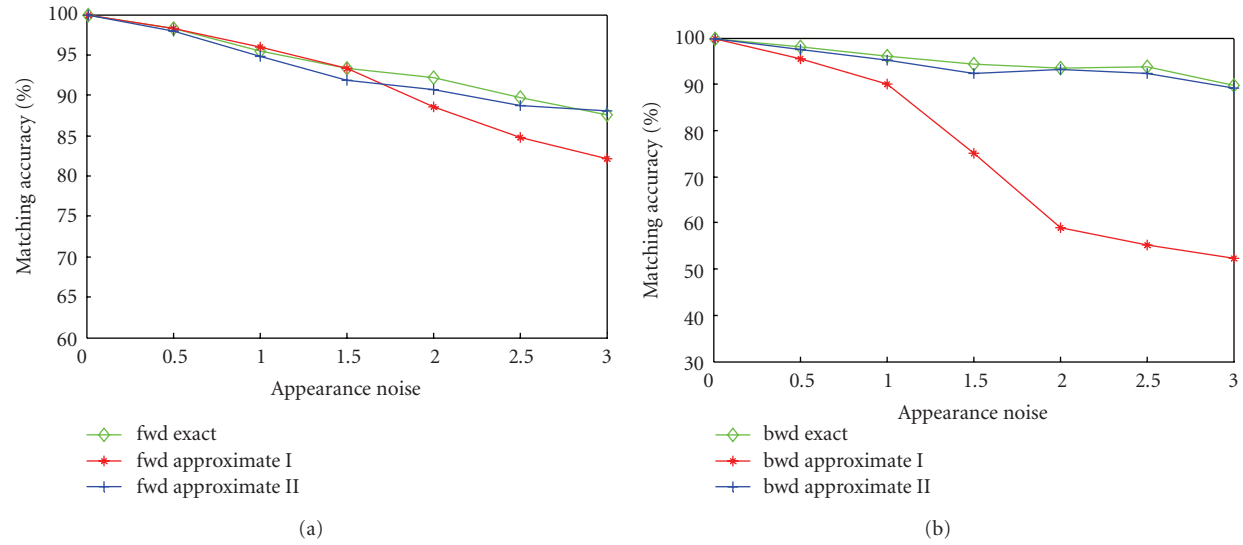

Figure 9: Mean accuracy under different appearance noise level. X-axis corresponds to the variance of appearance observations. (a) Forward inference. (b) Backward inference.

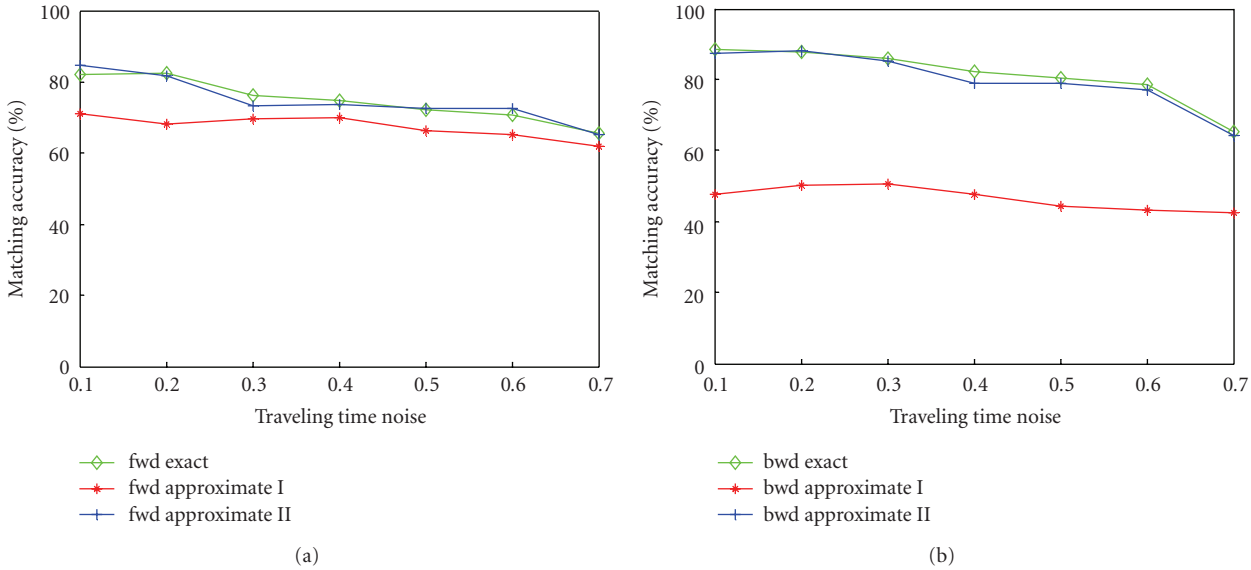

Figure 10: Mean accuracy under different level of traveling time standard deviation. X-axis corresponds to the ratio of standard deviation to mean value of the traveling time observations. (a) Forward inference. (b) Backward inference.

the standard deviation of travel time is assumed to be proportional to its mean value. In each simulation, for the kth object, we choose its starting node randomly in Figure 1, then generate its moving trajectory according to the transition probabilities π_{uv}. On each node along the trajectory of the kth object, the spatiotemporal observations d_i and appearance observations o_i are drawn from the assumed Gaussian model. We assume that at each time instance, there is only one object being observed by a camera. The observations of all objects are collected together and reordered according to the time observation, resulting in the data set $\{y_i\}$.

7.1.2. Evaluation Criteria. The criterion we use is the data association *accuracy*, denoted as q:

$$q = \frac{1}{K}\sum_{k=1}^{K} q_k \quad q_k = \frac{\left|\overline{Y}_k \cap Y_k\right|}{|Y_k|} \cdot 100\%, \tag{35}$$

where K is the number of objects of interest and $|\cdot|$ indicates the number of elements in a set. The term $\overline{Y}_k$ indicates the "ground truth" set of observations of object k, and Y_k is the set of observations of object k determined by the data association algorithms. To evaluate the complexity of

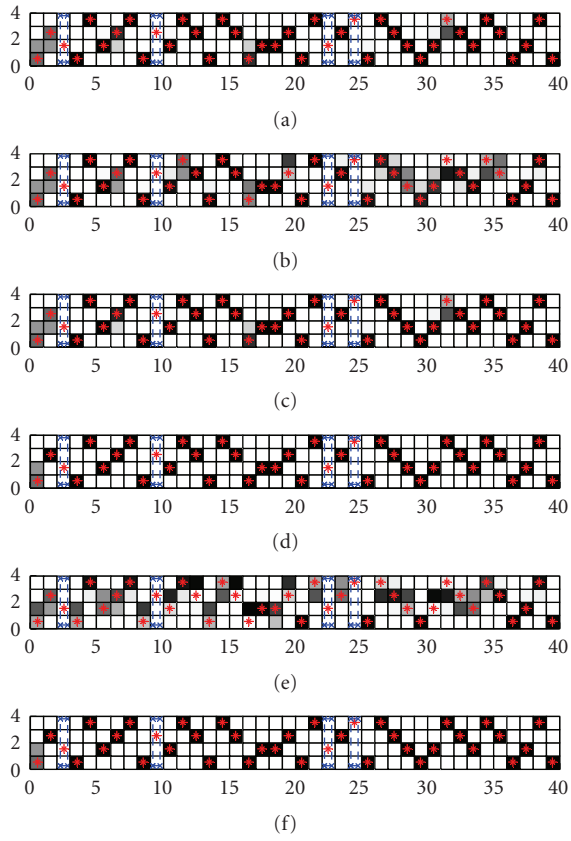

Figure 11: Marginal distribution of labeling variable in inference with 1-order spatiotemporal model. The 3rd, 10th, 23rd, and 25th observations are missed, depicted as dashed column. The true labels are depicted by stars. Each column represents the marginal distribution of the label of an observation. Grayscale corresponds to probability value. Black represents probability 1, and white probability 0. (a) Forward exact. (b) Forward approximate I. (c) Forward approximate II. (d) Backward exact. (e) Backward approximate I. (f) Backward approximate II.

Table 1: The mean travel time between adjacent nodes.

	a	b	c	d	e	f	g	h	i	j
a	0	58	80	85	0	0	0	0	0	0
b	58	0	0	0	77	97	0	0	0	0
c	80	0	0	61	46	60	93	0	0	0
d	85	0	61	0	0	0	44	42	0	0
e	0	77	46	0	0	79	0	0	48	0
f	0	97	60	0	79	0	84	0	66	0
g	0	0	93	44	0	84	0	61	0	94
h	0	0	0	42	0	0	61	0	0	71
i	0	0	0	0	48	66	0	0	0	0
j	0	0	0	0	0	0	94	71	0	0

algorithms, we simply use the running time of the Matlab implementation on a 1 GHz desktop PC.

7.1.3. Effect of Higher-Order Spatiotemporal Model in Case of Missing Detection. To examine the effect of spatiotemporal model (7) described in Section 3.3 in the case of missing detection, we compare the performance of data association using exact inference under different missing detection rates and spatiotemporal model orders. Note that the zero-order model is equivalent to the original spatiotemporal model (4). We first generate a data set consisting of 40 observations from 4 objects, delete certain number of observations from it randomly, and then apply on it the exact forward and backward inference algorithms described in Section 4. The process is repeated 200 times, and the mean data association accuracy is shown in Table 1.

It can be seen from Table 1 that as the number of missed observations increases, the accuracy of 0-order model-based inference algorithm decreases obviously. And the performance of backward inference is even worse than that of the forward. When the missing detection rate is high, for example, 4 or 8 missed, the 2-order model gives the best results. In our simulation we note that the model with order higher than 3 does not give better performance. This may be due to the following. (i) The consecutively missing detection of a single object rarely occurs in the simulations; (ii) the higher the model order, the higher the variance of traveling time; (iii) as shown in (11), the higher-order model weakens the effect of spatiotemporal observation by distributing the information to multiple path according to the coefficients $p(\omega_i^{(u,v)})$. If other information indicating the path taken by the object is available, such as the entry or exit direction, the performance of higher-order model-based inference may be improved. The additional computational burden introduced by higher-order model is the construction of the composite parameter matrix $\overline{A}$, which only needs to be calculated once. The difference of running times of inference algorithms based on different order model is negligible. It should be noticed that here our focus is on the effect of high-order model on missing detection. Hence in Table 2 we only list the results using exact inference. The results given by approximate inference are similar, as shown in the following discussions.

Figure 6 shows the marginal distributions of the labeling variable in forward and backward pass in a sample run. It can be seen clearly in Figure 6(a) that, after the missing detections occurred in steps 24 and 34, a large number of observations have been mislabeled in the following steps, resulting in a low association accuracy of 73.61% in the forward inference with 0-order model. In contrast, the forward pass with the 1-order spatiotemporal model gives a perfect association after step 24, as shown in Figure 6(b), improving the accuracy to 92.73%. We note that, in the backward pass with 0-order model as shown in Figure 6(c), the number of mislabeled observations increases compared with that in Figure 6(a), resulting in the accuracy of 41.67%. This may be attributed to the further mistakes introduced by the backward inference on the incomplete data. However, it is shown in Figure 6(d) that the backward inference with 1-order model achieves a 100% correct association.

In Figure 7 we compare the marginal distributions of the labeling variables in exact forward and backward inference with spatiotemporal model of different orders. Note that object 2 is miss detected consecutively at steps 8 and 13

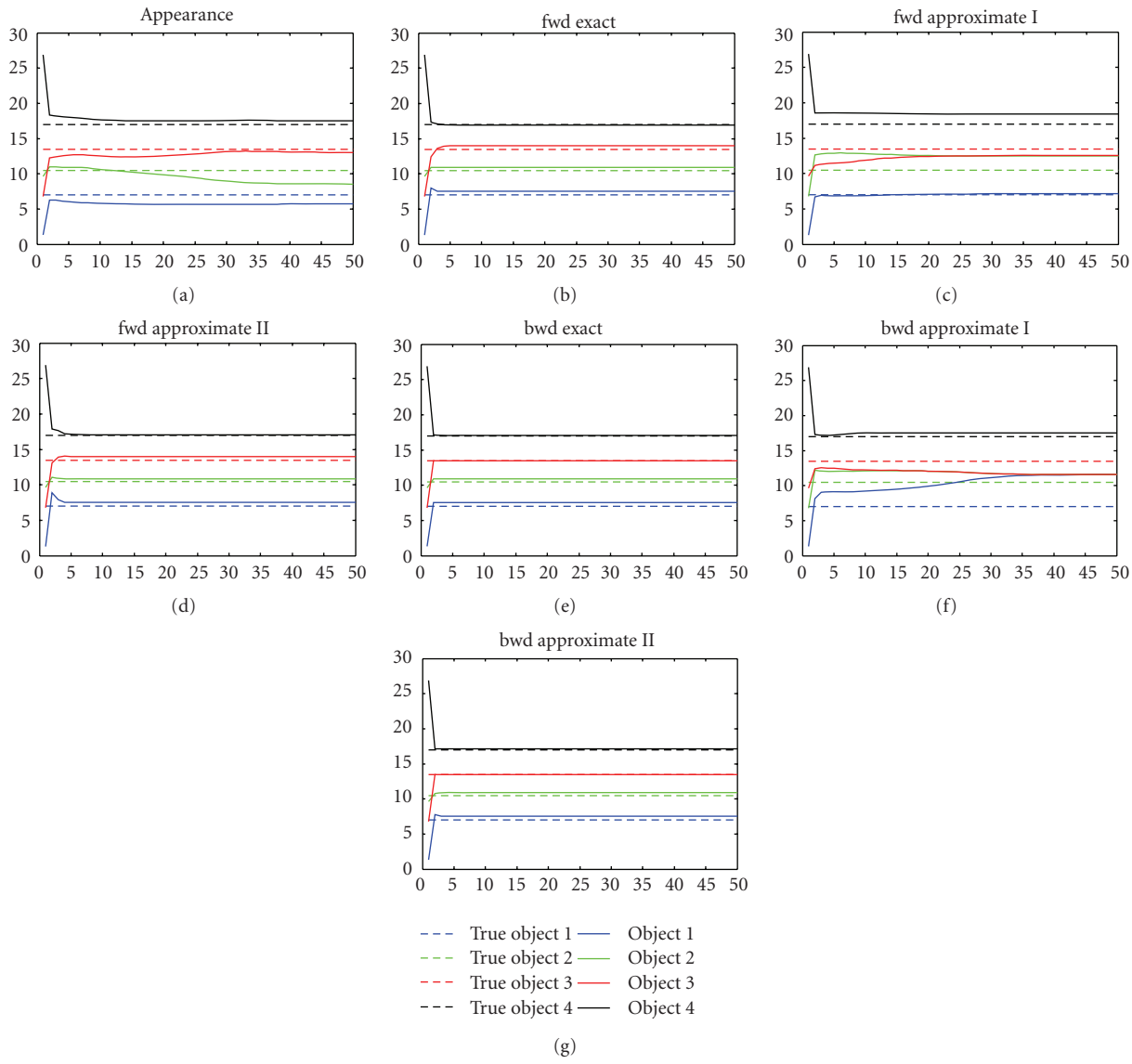

Figure 12: Parameter learning curve of EM with different inference algorithms. (a) Standard EM. (b) Forward exact. (c) Forward appr.I. (d) Forward appr.II. (e) Backward exact. (f) Backward appr.I. (f) Backward appr.II.

Table 2: Means of the accuracy of exact inference in case of missing detection (%).

Missed obs.	1		2		4		8	
	Forward	Backward	Forward	Backward	Forward	Backward	Forward	Backward
0-order	87.37	85.01	85.00	81.26	81.69	74.91	75.55	64.84
1-order	92.49	95.65	90.51	95.19	85.18	87.12	82.15	86.02
2-order	91.16	94.85	89.43	94.56	88.13	93.23	82.18	85.44
3-order	88.38	91.24	85.69	89.44	83.99	83.02	78.93	72.60

and observations 32 and 33 are missed consecutively. Data association using 0-order model results in the accuracy of 63.45% and 80.56% in forward and backward pass, respectively, as shown in Figures 7(a) and 7(d). Using 1-order model can improve the association accuracy to 69.42% and 90.97% in forward and backward pass respectively. However, due to the consecutive missing detection, there are still a large number of mislabeled observations in Figures 7(b) and 7(e). The best results are achieved by the 2-order model-based inference, as shown in Figures 7(c) and 7(f), with association accuracy of 71.86% and 90.97% in forward and backward pass, respectively. The results show that the 2-order model can improve the robustness of the inference algorithms against consecutive missing detections.

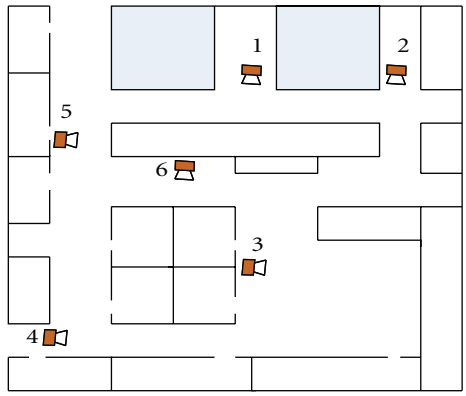

Figure 13: Experiment setup. Building plan where the observations were taken.

7.1.4. Exact versus Approximate Inference. In this subsection we compare the performance of inference algorithms described in Sections 4 and 5. Firstly, we compare the inference errors of *approximate inference I* and *approximate inference II* (we denote them as appr.I and appr.II in the following discussion). As before, we denote the marginal distribution of the labeling variable calculated with exact inference as $p(x_i)$ (including the forward pass marginal $p(x_i \mid y_{0:i})$ and the backward marginal $p(x_i \mid y_{0:N})$) and denote the marginal distributions calculated with appr.I and appr.II as $\hat{p}(x_i)$ and $\widehat{p'}(x_i)$, respectively. We use the Kullback-Leibler divergence to measure the discrepancy between $p(x_i)$ and $\hat{p}(x_i)$, and that between $p(x_i)$ and $\widehat{p'}(x_i)$. The KL divergence is calculated as

$$D[p(x_i) \,||\, \hat{p}(x_i)] \triangleq E_p\left[\ln \frac{p(x_i)}{\hat{p}(x_i)}\right] = \sum_{x_i} p(x_i) \ln \frac{p(x_i)}{\hat{p}(x_i)}. \tag{36}$$

Figure 6 shows the KL divergence caused by approximate inference in forward and backward pass. We run the algorithms on the data set of 100 observations. The simulation is repeated 1000 times. The KL divergence data in each simulation are recorded, and 10 of them are concatenated into a long vector, which is depicted in Figure 8.

Form Figure 8 we observe that the appr.I inference has a much higher KL divergence, averaging at 0.3933 for forward pass and 1.1727 for backward, than that of the appr.II inference, averaging at 0.0127 for forward pass and 0.0129 for backward. Moreover, the KL divergence of appr.I inference has much more spikes than that of appr.II, both in forward and backward pass. However, it is shown that neither the error of appr.I nor that of appr.II appears to grow over the length of run.

To examine the robustness of inference algorithms against appearance and traveling time observation noise, we compare the association accuracy between the exact and approximate inference algorithms under different noise level and depict the results in Figures 9 and 10. The statistics under each noise level is summarized from the results of 200 sample runs on observation sets generated from 4 objects. Observations are deleted randomly according to the missing detection rate. Figure 9 depicts the behavior of the mean accuracy under different appearance variation, where the standard deviation of traveling time is set to 10% of the mean value. Figure 8 depicts the behavior of the mean accuracy under different traveling time noise, where the variance of appearance is set to 4.

It can be seen from Figure 9 that the accuracy of inference algorithms decreases with the increasing appearance noise level. However, as shown in Figure 9, the accuracy of appr.I inference drops much faster than that of the exact and appr.II inference when the appearance noise increases. The accuracy of backward appr.I inference drops very fast and is equal to 52.28% when the appearance variance increases to 3. It is shown in Figure 10 that the accuracy of inference algorithms decreases with the increasing noise of traveling time. When the standard deviation increases to 70% of the mean value, the accuracy of forward exact and appr.II inference drops to 65.68% and 65.12% and that of the backward exact and appr.II inference drops to 65.32% and 64.10%, respectively. In Figures 9 and 10, it is clear that the performance of appr.I inference is obviously inferior to the other two methods and the performance of appr. II inference is comparable to that of the exact inference in terms of association accuracy. But the former is much faster than the latter, as shown in Table 4. The results shown in Figures 9 and 10 are consistent with the KL divergence analysis above.

Figure 11 shows the marginal distributions of the labeling variable in a sample run of different forward and backward inference algorithms with 1-order spatiotemporal model. We can see that the label's marginal distributions given by appr.II inference, shown in Figures 11(c) and 11(f), are almost the same as these given by the exact inference, shown in Figures 11(a) and 11(c). In this sample run, the appr.II inference has the same labeling accuracy as that of exact, 97.50% in forward pass, and 100% in backward pass, respectively. On the other hand, due to the larger distribution representation error in appr.I inference, the resulting marginals, shown in Figures 11(b) and 11(e) are much inconsistence with those given by exact inference. In this sample run, the forward appr.I inference has an accuracy of 86.11%, and the backward is even worse, of 62.41%.

To illustrate the scaling property of the inference algorithms we record the performance of algorithms on data sets of different scales. The rate of missing detection is 10%, and the variance of appearance observation is 2. The results are shown in Tables 3-4.

From Table 3 we can see that, except MCMC, the accuracy of each inferences algorithm is consistent on data sets of varying scales. The exact and appr.II inference give the best results. However, the computational burden of exact inference grows exponentially as the data increase, rendering it inexcusable due to the memory limitation when the data set contains 4×20 observations or more. The appr.I inference is very fast, but its accuracy in backward pass is unacceptable. The appr.II inference is slower than appr.I, but it is still much faster than exact inference, achieving a better compromise between computational simplicity and inference accuracy.

Figure 14: True trajectories of persons. Missing detections are depicted by dashed boxes.

Table 3: Mean of the accuracy of inference algorithms under different number of observations (%).

No. obs.	MCMC	Exact		Approximate I		Approximate II	
		Forward	Backward	Forward	Backward	Forward	Backward
3×10	74.24	90.96	94.71	88.86	58.93	90.23	93.48
3×20	64.53	90.50	93.09	89.78	60.88	90.35	91.90
4×20	58.01	x	x	86.05	56.08	91.32	93.74
5×20	44.28	x	x	83.41	56.02	89.60	92.51

7.1.5. Comparison with MCMC Method. We also compare the proposed algorithms with Markov Chain Monte Carlo method, which is widely used in data association problems [1, 15, 16]. MCMC is a sampling method for finding a good approximate solution to a complex distribution. It draws samples ω from a distribution π on a space Ω by constructing a Markov chain. The transition in Markov chain may be set up in many ways as long as ergodicity is ensured. We use Metropolis-Hastings algorithm for MCMC sampling, where a transition from ω to ω' follows the proposal distribution $q(\omega'|\omega)$ and the move is accepted with the acceptance probability

$$\alpha = \min\left(1, \frac{\pi(\omega')q(\omega \mid \omega')}{\pi(\omega)q(\omega' \mid \omega)}\right). \tag{37}$$

In our application, the sample space Ω consists of all possible partitions of the observation set Y into a fixed number of mutually exclusive subsets $\{Y_k\}$, such that each subset Y_k contains all observations believed to come from a single object. The stationary distribution π is the posterior $p(\omega|Y)$. The transition from ω to ω' is implemented with update move; please refer to [7] for details. In each simulation, we run the MCMC 10 times independently with randomly chosen initial sample and then find the partition ω with the maximum posterior probability. The number of samples in each MCMC run is set to 10^4.

The original MCMC method is unsuitable in case of missing detection. For fair comparison we implement MCMC which use the 1-order spatiotemporal model (7) in the evaluation of posterior of each trajectory instead. The accuracy and running time of MCMC are shown in Tables 3 and 4. It can be seen that the accuracy of MCMC-based data association algorithm is lower than that of the inference algorithms presented in this paper. In our simulations, we note that MCMC method is unsuitable for recovering long trajectories. As shown in Table 3, the association accuracy of MCMC drops rapidly when the number of observations of each object increases. In contrast, our methods show consistent performance on data sets of varying scale. Moreover, the running time of MCMC is longer due to the large number of samples needed to be generated to cover the sample space.

7.1.6. Inference with Unknown Parameters. We also study the performance of the proposed inference algorithms in EM framework when the prior probability of the label of observations α_k, the mean and variance of appearance μ_k and σ_k are unknown. Setting the mean of appearance as [7, 10.5, 13.5, 17] and the variance as 2, we generate the observation set of 4 objects. Firstly we use the standard EM for GMM model [31] to learn the parameters and determine the hidden label of each observation. The ownership probability of each observation in standard EM is calculated as

$$p(x_i = k \mid o_i, \Theta) = \frac{N\left(o_i; \mu_k, \sigma_k^2\right)}{\sum_j N\left(o_i; \mu_j, \sigma_j^2\right)}. \tag{38}$$

In standard EM only appearance information is used, and observations are assumed to be mutually independent. To exploit the spatiotemporal information, we replace (38) with different inference algorithms presented before. With random initialization, the parameter learning curves of

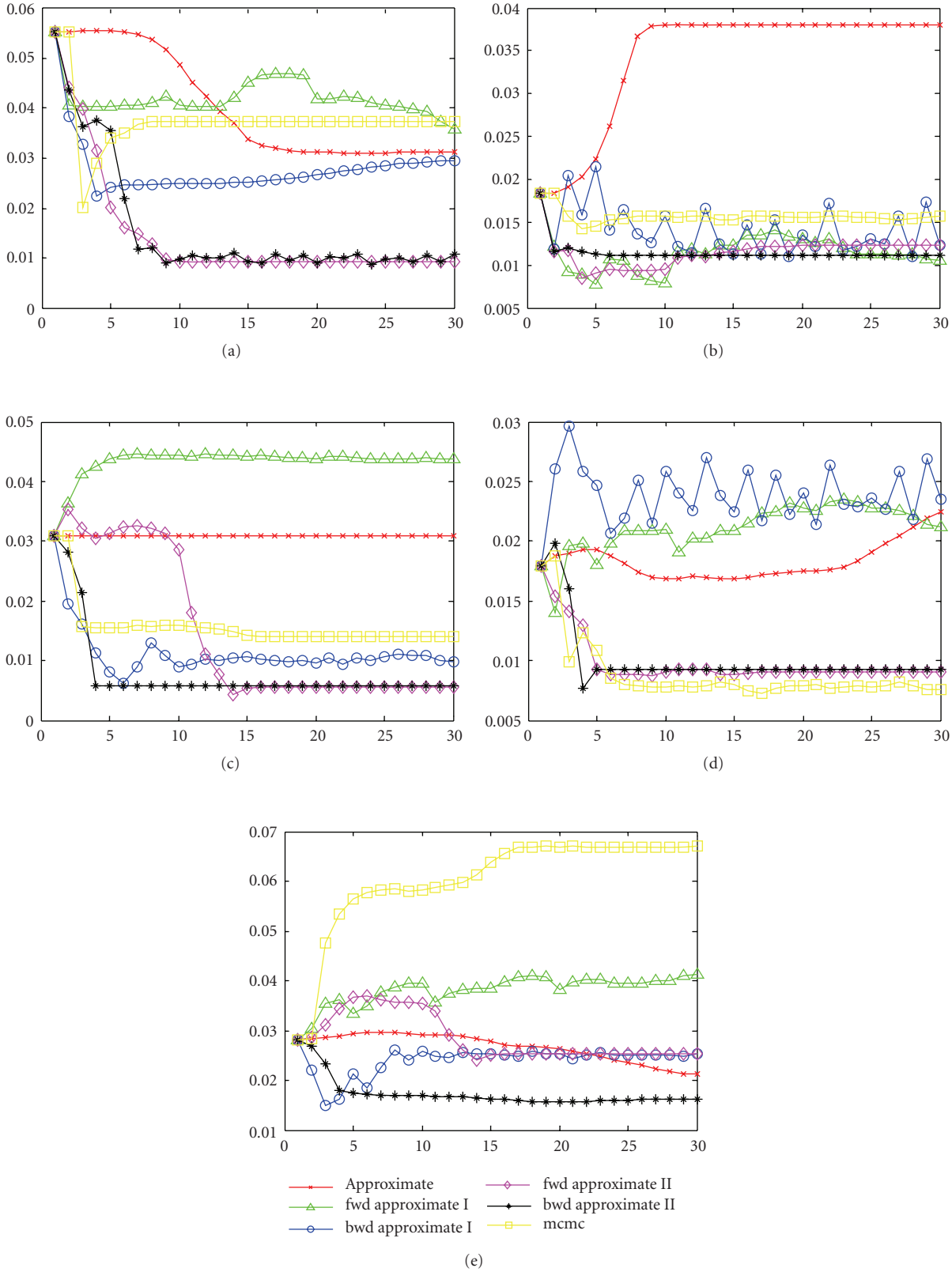

Figure 15: Learning curve of EM with different inference algorithms. X-axis corresponds to the index of EM iterations; Y-axis corresponds to the Euclidean distance in the channel normalized space [32] between the estimated and true appearance mean. (a)–(e) corresponds to person A to E, respectively.

Figure 16: Trajectories recovered by standard EM initialized by K-means clustering.

Figure 17: Trajectories recovered by EM with MCMC inference initialized by K-means clustering.

Figure 18: Trajectories recovered by EM with forward appr.I inference initialized by K-means clustering.

Figure 19: Trajectories recovered by EM with backward appr.I inference initialized by K-means clustering.

Figure 20: Trajectories recovered by EM with forward appr.II inference initialized by K-means clustering.

Figure 21: Trajectories recovered by EM with backward appr.II inference initialized by K-means clustering.

Table 4: Running time of inference algorithms under different number of observations (s).

No. obs.	MCMC	Exact		Approximate I		Approximate II	
		Forward	Backward	Forward	Backward	Forward	Backward
3×10	185.24	4.9809	8.5448	0.1066	0.4366	1.3130	1.9201
3×20	335.67	73.4384	131.81	1.3776	2.7045	8.2304	10.633
4×20	410.36	x	x	7.9897	13.284	41.750	51.743
5×20	455.80	x	x	29.992	46.791	149.49	181.99

Table 5: Mean of data association accuracy of different inference algorithms in EM framework (%).

No. obs.	Standard EM	Exact		Approximate I		Approximate II	
		Forward	Backward	Forward	Backward	Forward	Backward
3×10	63.28	84.37	91.91	72.21	61.22	78.79	88.82
3×20	59.02	81.78	89.93	70.86	52.62	80.38	86.72
4×20	53.82	x	x	70.07	57.58	82.66	87.16
5×20	57.04	x	x	65.58	57.78	85.69	87.71

standard EM and EM using different inference engines are shown in Figure 12.

As shown in Figure 12, in this sample run, the learning curves of EM with exact inference and EM with appr.II inference converge to the true parameter values. However, the learning curves of standard EM and EM with appr.I inference are stuck in local maximums. In simulations we find that the EM with exact inference and appr.II inference is more likely to converge to the true parameter value than the standard EM and EM with appr.I inference. This suggests that the effective use of spatiotemporal information can improve the robustness of EM against the problem of local traps.

Table 5 shows the mean accuracy of data association of different inference algorithms in EM framework on data sets of various scales. The statistics are obtained from 200 sample runs. Comparing Tables 5 and 3 we find that the mean accuracy of appr.I inference drops significantly. We observe that, in some simulation runs, the parameter estimate of EM with appr.I inference does not converge to the true value, thus resulting in low association accuracy.

7.2. Application on Multiple Persons Tracking with VSN

7.2.1. Setup. We test the presented methods on real world human observations that were collected by cameras at 6 disjoint locations in an office building. The building plan and the corresponding topological model of VSN are shown in Figure 13.

In total we gather 75 observations of 5 persons, with an equal number of observations per person. Each observation consists of the appearance feature of the captured person, the median time of the person's present in front of the camera, and the moving direction of the person in the camera's FOV. For this observation set we manually resolve the data association to obtain the "ground truth" partition, as shown in Figure 14. In this data set, we delete randomly chosen 10 from the total 75 observations, which are depicted by dashed boxed in the figure. Note that consecutive missing detections occurred in the trajectory of person E.

7.2.2. Observations. The appearance feature summarizes person appearance information contained in a sequence of frames during the person's presence in a camera field of view. To extract the appearance feature, we first manually segment the interested person from the frames and then compute the color means (RGB) over three regions of the person images. The regions are selected heuristically as in [18], and the resulting features provide a simple way to summarize color while preserving partial information about geometrical layout. Thus the appearance feature in each observation is a 9D vector. To suppress the effect of the illumination, we transform the original RGB representation to a channel normalized space [32].

In practice, the walking speeds of different persons may be quite different. Moreover, occasional stops may occur during person's moving from one camera to another. These factors increase the variance in the spatiotemporal model and weaken the discriminative power of traveling time measurement. To overcome this difficulty, the moving direction of persons in the camera's FOV can be used as additional spatiotemporal feature. The moving direction features are represented by the borders via which the person arrives to and departs from the camera's FOV [18]. The modified spatiotemporal model can be easily incorporated into the Bayesian inference framework.

7.2.3. Evaluation Criteria. For a good multiobject tracking or trajectory recovering algorithm, it is desirable that [18] (i) observations assigned to the same trajectory correspond to a single object; (ii) all observations of a single object belong to the same trajectory. Correspondingly, we use the following three criteria to evaluate various algorithms.
The *precision*

$$P = \frac{1}{K}\sum_{s=1}^{K}\frac{\max_i \left|\widehat{C}_s \cap C_i\right|}{\left|\widehat{C}_s\right|}. \tag{39}$$

The *recall*

$$R = \frac{1}{K}\sum_{i=1}^{K}\frac{\max_s \left|\widehat{C}_s \cap C_i\right|}{|C_i|}. \tag{40}$$

The *F1-measure*

$$F1 = \frac{2 * P * R}{P + R}, \tag{41}$$

where K is the number of objects under tracking and $|\cdot|$ indicates the number of elements in a set. The term C_i indicates the "ground truth" trajectory of object i, and $\widehat{C}_s$ is the sth trajectory generated by the tracking algorithms. Note that *F1-measure* is the harmonic mean of the *precision* and *recall*.

7.2.4. Experimental Results. Firstly, we apply K-means clustering on the observations set shown in Figure 14 to obtain

Table 6: Data association accuracy of difference algorithms (%).

	Fixed appearance model obtained by K-means clustering			Adaptive appearance model using EM initialized with K-means clustering		
	Precision	Recall	F1	Precision	Recall	F1
App.	65.69	55.56	60.21	63.07	55.23	58.89
MCMC	59.21	63.10	61.09	64.67	53.90	58.79
Fwd appr.I	72.87	65.10	68.77	71.60	57.56	63.82
Bwd appr.I	59.85	51.18	55.18	67.11	67.85	67.48
Fwd appr.II	65.03	59.10	61.92	88.46	88.46	88.46
Bwd appr.II	68.92	60.90	64.66	94.29	92.33	93.30

a rough estimate of the mean and covariance of each person's appearance. Based on the obtained appearance parameters, different inference algorithms are used for trajectory recovering. It can be seen from Table 6 that, using the fixed appearance model given by K-means clustering, none of inference algorithms can give satisfactory result. However, if we use the K-means clustering results as initial value and update the appearance parameters using EM with different inference, the performance can be improved, especially in the case of approximate inference II. The results in Table 6 clearly demonstrate the power of the combination of EM and spatiotemporal based inference.

Figure 15 shows the behavior of the Euclidean distance in the channel normalized space [32] between the estimated and true appearance mean of the five persons during EM iterations using different inference algorithms. The distance at iteration t is calculated as

$$d_k(t) = \sqrt{(\hat{\mu}_k(t) - \mu_k)'(\hat{\mu}_k(t) - \mu_k)}, \tag{42}$$

where μ_k and $\hat{\mu}_k(t)$ are the true and estimated appearance mean of the kth person, respectively. It can be seen form Figure 15 that the EM using approximate inference II can always achieve the most accurate estimate of the appearance parameters rapidly. In contrast, the EM using appearance-based inference, EM using approximate inference I and EM using MCMC inference, may result in an estimate even worse than that given by K-means clustering.

Figures 16–21 show the trajectories recovered by EM with appearance based inference, EM with MCMC inference, EM with forward approximate inference I, EM with backward approximate inference I, EM with forward approximate inference II, and EM with backward approximate inference II, respectively. It can be seen form Figure 16 that due to the varying observing condition, the appearance of the same person changes significantly at different cameras and the algorithm based solely on appearance information gives a very poor recovering performance. Although the spatiotemporal information is used, the recovering performance is still unsatisfactory due to the inaccuracy in MCMC and approximate inference I, as shown in Figures 17–19. In contrast, Figures 20 and 21 show that EM with approximate inference II can improve the recovering performance significantly. Note that the trajectory of person D is recovered perfectly in Figures 20 and 21 and trajectories of person A are recovered perfectly in Figure 21. The recovered trajectories of persons B, C, and E show the effects of higher-order spatiotemporal model (we use 2-order model) in case of missing detection.

8. Conclusions

In this paper we address the problem of data association in visual sensor networks. We consider data association as an inference problem in dynamic Bayesian networks, where a higher-order spatiotemporal model is used to describe the probabilistic dependency between observations. As the exact inference on DBN is intractable, we present two kinds of approximation schemes of the exact belief state and derive the corresponding forward and backward inference algorithms. Finally we incorporate the proposed model and algorithms into EM frameworks to account for the unavailability of prior knowledge about object appearance. Simulation and experimental results show that the higher-order spatiotemporal model leads to improved association accuracy in case of missing detection. The approximation inference algorithms are much faster than exact inference in case of large scale data set, and the inference algorithm based on the second approximation schemes has better performance in terms of association accuracy.

There are two interesting directions deserving further investigation. First, in our method the number of objects under tracking is assumed to be known *a priori*. However, in many applications this is not true. In this case, the observation set should be explained with an infinite mixture model, the parameters of which can be estimated using the theory of Dirichlet process [33]. Second, the proposed method is a centralized approach in that it needs to collect all data into a data processing center. This is unsuitable for large-scale visual sensor networks. Nowadays, smart camera emerges which is not only able to capture videos, but also to memorize and process the information and communicate with each other [34]. It is desirable that global data association is achieved through the local information processing on each camera nodes and the information exchange between them. We are working in theses directions.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities of China and Beijing Natural Science Foundation no. 4113072. The authors would like to

thank the student volunteers for their participation in the tracking experiments presented in this paper. The authors are grateful to the anonymous reviewers for their valuable suggestions for improving the quality of the paper.

References

[1] S. W. Yeom, T. Kirubarajan, and Y. Bar-Shalom, "Track segment association, fine-step IMM and initialization with doppler for improved track performance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 1, pp. 293–309, 2004.

[2] A. Gilbert and R. Bowden, "Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity," in *Proceedings of the 9th European Conference on Computer Vision (ECCV '06)*, vol. 3952, pp. 125–136, 2006.

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

[4] H. Pasula, S. Russell, and M. Ostland, "Tracking many objects with many sensors," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1160–1171, 1999.

[5] W. Zajdel and B. Klose, "Gaussian mixture models for multi-sensor tracking," in *Proceedings of the Dutch-Belgian Artificial Intelligence Conference*, pp. 371–378, 2003.

[6] W. Zajdel and B. J. A. Kröse, "A sequential bayesian algorithm for surveillance with nonoverlapping cameras," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 8, pp. 977–996, 2005.

[7] H. Kim, J. Romberg, and W. Wolf, "Multi-camera tracking on a graph using Markov chain Monte Carlo," in *Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '09)*, September 2009.

[8] F. Van De Camp, K. Bernardin, and R. Stiefelhagen, "Person tracking in camera networks using graph-based Bayesian inference," in *Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '09)*, September 2009.

[9] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*, John Wiley & Sons, New York, NY, USA, 2001.

[10] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[11] P. Willett, Y. Ruan, and R. Streit, "PMHT: problems and some solutions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 738–754, 2002.

[12] J. A. Roecker and G. I. Phillis, "Suboptimal joint probabilistic data assoication," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 2, pp. 510–572, 1993.

[13] X. Wang and D. Mušicki, "Low elevation sea-surface target tracking using IPDA type filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 2, pp. 759–774, 2007.

[14] S. Godsill and J. Vermaak, "Variable rate particle filters for tracking applications," in *Proceedings of the IEEE/SP 13th Workshop on Statistical Signal Processing*, pp. 1280–1285, July 2005.

[15] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for multi-target tracking," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 481–497, 2009.

[16] Y. Goyat, T. Chateau, and F. Bardet, "Vehicle trajectory estimation using spatio-temporal MCMC," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, Article ID 712854, 9 pages, 2010.

[17] X. Boyen and D. Koller, "Tractable inference for complex process," in *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, 1998.

[18] W. Zajdel, *Bayesian visual surveillance*, Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands, 2006.

[19] C. C. Loy, T. Xiang, and S. Gong, "Multi-camera activity correlation analysis," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 1988–1995, June 2009.

[20] B. Song and A. K. Roy-Chowdhury, "Robust tracking in a camera network: a multi-objective optimization framework," *IEEE Journal on Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 582–596, 2008.

[21] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 253–259, June 1999.

[22] R. Farrell, D. Doermann, and L. S. Davis, "Learning higher-order transition models in medium-scale camera networks," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, October 2007.

[23] G. Casella and R. L. Berger, *Statistical Inference*, Wadsworth Group, 2002.

[24] K. Murphy, *Dynamic Bayesian networks: representation, inference and learning*, Ph.D. thesis, University of California, Berkeley, Berkeley, Calif, USA, 2002.

[25] X. Boyen and D. Koller, "Exploiting the architecture of dynamic systems," in *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pp. 313–320, July 1999.

[26] R. Shachter, "Bayes-Ball: the rational pastime for determining irrelevance and requisite information in belief networks and influence diagrams," in *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, 1998.

[27] A. Globerson and T. Jaakkola, "Approximate inference using conditional entropy decomposition," in *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, 2007.

[28] K. Murphy and Y. Weiss, "The factored frontier algorithm for approximate inference in DBNs," in *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, 2001.

[29] F. Hutter, N. Brenda, and R. Dearden, "Incremental thin junction trees for dynamic Bayesian networks," Tech. Rep. TR-AIDA-04-01, Intellectics Group, Darmstadt University of Technology, 2004.

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, pp. 1–38, 1977.

[31] J. Bilmes, "A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Tech. Rep. ICSI-TR-97-021, University of California, Berkeley, Berkeley, Calif, USA, 1997.

[32] M. S. Drew, J. Wei, and Z. N. Li, "Illumination-invariant color object recognition via compressed chromaticity histograms of

color-channel-normalized images," in *Proceedings of the IEEE 6th International Conference on Computer Vision*, pp. 533–540, January 1998.

[33] M. Beal, Z. Ghahramani, and C. Rasmussen, "The infinite hidden Markov model," in *Proceedings of the Advances in Neural Information Processing System*, 2002.

[34] B. Rinner and W. Wolf, "A bright future for distributed smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1562–1564, 2008.