Research Article

Distributed Bayesian Inference for Consistent Labeling of Tracked Objects in Nonoverlapping Camera Networks

Jiuqing Wan and Li Liu

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

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

Received 4 June 2013; Revised 30 September 2013; Accepted 21 October 2013

1. Introduction

Recently, there has been increasing research interest in wide-area video surveillance based on smart camera networks with nonoverlapping Field of View (FOV). The cameras in the networks are not only able to capture video data, but also capable of local processing and mutual communication. They usually work cooperatively for discovering and understanding the behavior of some interested objects, for example, pedestrians and vehicles, in the monitored region. One of the fundamental prerequisites for achieving this goal is the correct labeling of the observations of objects captured by each camera node in a consistent way. That is, the observations assigned with the same label are assumed to be generated from the same object. In this paper, we assume that the detection and tracking problem within a single camera view has been already solved, and we call some interesting quantities extracted from the tracked object as a virtual “observation;” see Figure 2.

Consistent labeling of tracked objects in nonoverlapping camera networks is, however, a rather challenging task. First, the object’s appearance often undergoes large variations across disjoint camera views due to significant changes in view angle, lighting, background clutter, and occlusion. The different objects may appear more alike than that of the same object across different views. This makes the labeling of objects based solely on appearance cues to be very difficult. Second, for large scale application, it is unrealistic to transmit all video data collected by cameras in the networks to a central server for labeling inference due to the limitation of communication bandwidth and camera node energy. Even through smart camera can analyse the local video and transmit only extracted features, the central server will become overwhelmed quickly when the number of objects and measurements increases because of the combinatorial nature of labeling problem. Thus, distributed algorithms are preferred rather than centralized one. Third, the uncertainty in the number of objects presenting in the monitored region makes the labeling problem even more difficult, as we should infer not only the label for each tracked object but also the...
number of possible labels, that is, how many objects are moving in the region, at the same time.

1.1. Related Works. A lot of works have been proposed to answer the above challenges. In the community of object reidentification research, huge amounts of efforts [3–8] are devoted to matching objects across different views based on the unreliable appearance measurements. However, the results are still far from satisfactory and the outputs of reidentification algorithms are always in the form of a list of top ranked candidates, which can not be used directly for consistent labeling. Recently, some works have been shown [9] in which spatiotemporal cues, such as the capturing time and moving direction, are exploited to improve matching accuracy. But in [9], the authors only consider matching problem between a pair of views, and extending it to camera networks is not a trivial matter.

The problem of consistent labeling in camera networks using both appearance and spatiotemporal cues has been widely investigated, under the name of data association [1, 10], trajectory recovery [11], or camera-to-camera tracking [12–17]. Some authors try to solve the problem by optimally partitioning the set of observations collected by the camera networks into several disjoint subsets, such that the observations in each subset are believed to come from a single object. The difficulty caused by the exponential growth of the partition space is tackled by making appropriate independence assumptions and leveraging efficient optimization algorithms, such as Markov Chain Monte Carlo [11, 14], Max-Flow Network [15], or Multiple Hypothesis Tracking [16, 17]. On the other hand, it is more attractive to treat the problem in a Bayesian framework [1, 10], in which each observation is assigned with a labeling variable, and the posterior distribution of the labeling variable is inferred, conditioned on all evidence made in whole networks. The resulting marginal distribution of labeling variable contains the complete knowledge about which object the observation has originated from. However, doing inference in the joint labeling space is usually intractable and the authors have to resort to some assumed independence structure and approximate algorithms such as Assumed Density Filter [1, 10].

The above approaches are all centralized, making them not suitable for using in large-scale camera networks for reasons mentioned before. Recently, distributed solutions of consistent labeling problem, which involves only local information processing and exchanging while being able to achieve the same or similar labeling accuracy as their centralized counterparts, have attracted many research interests. Considering the appearance observations made in the networks as i.i.d. samples drawn from a mixture model, and treating the labeling variables as missing data, various appearance-based distributed labeling methods have been proposed under framework of distributed EM [18–21]. However, these algorithms always perform poorly when observing conditions vary largely across camera views, as they assume that the appearance of a single object follows a unimodal distribution. To improve the labeling performance, exploiting the spatiotemporal information is necessary. Unfortunately, unlike the case of traditional wireless sensor networks [22–28] or camera networks with overlapping FOVs [29–31], where the dependence of involved variables in spatial-dimension (intrascan dependence) and temporal-dimension (intrascan dependence) can be modeled separately, the spatial and temporal evidence made in nonoverlapping camera networks are tightly coupled. This precludes the use of most existing distributed inference or optimization algorithms for traditional WSN and overlapping camera networks. In our recent works [2], based on the nonmissing detection assumption, we use a spatiotemporal tree to model the dependence of involved variables, and use belief propagation algorithm for calculating the posterior probability of labeling variable, which can be viewed as observation ownership in E-step of the distributed EM framework. Compared with traditional distributed EM, significant performance gain has been obtained by the effective using of spatiotemporal information.

1.2. Our Contributions. The main limitations of the work in [2] are (i) the number of objects under tracking needs to be known beforehand and (ii) the appearance of a single object is assumed to follow a Gaussian distribution. In this paper, we propose a new distributed Bayesian inference framework for consistent labeling of the tracked objects in nonoverlapping camera networks, which nicely overcomes the above limitations.

In our method, under the same nonmissing detection assumption as in [2], the posterior distribution of each labeling variable conditioned on all appearance and spatiotemporal measurements made in the networks is calculated, based solely on local inference on each camera nodes and belief propagation between neighboring cameras. This is possible because the nonmissing detection assumption ensures that when the label of an observation made on a specific camera is inferred, all relevant information made by the networks have been summarized in the belief states of labeling variables corresponding to observations already generated on the camera’s neighbors. Here, neighboring cameras refer to cameras connected by an edge in the topology of camera networks, see Figure 1.

Unlike [2], we do not prespecify a fixed number of possible objects, that is, a fixed sampling space for each labeling variables. Instead, we allow each newly arriving observation to ignite a new possible object, or equivalently, to add a new element in the sampling space, as we notice the fact that each observation is either originated from a previously observed object or from a newly appeared one. Based on the nonmissing detection assumption mentioned before, the sampling space of current labeling variable is determined in an online and distributed manner on each camera, by combining the sampling spaces of labeling variables already generated on neighboring cameras. Through the propagation of sampling spaces, it can be shown that each camera always performs inference in a space consisting of identifiers of all possible objects that may produce the current observation.

In addition, in this work, we discard the Gaussian assumption of object appearance used in [2]. Instead, we only assume that two observations are “similar” if they originate
from the same object. Here, saying that two observations are similar in appearance implies higher rank in the output list of some appearance-based object reidentification algorithm such as [3–8]. Consequently, two observations of the same object that look quite different in original color space due to variance in illuminating conditions may get higher similarity score by leveraging advanced techniques developed in object reidentification community. Spatiotemporal similarity is determined by the level of fitting of the spatiotemporal part of two observations to the spatiotemporal model, which is learned from training data or prespecified according to the prior knowledge of monitored region. In our Bayesian frameworks, the likelihood of observation is defined in terms of above similarity measures between observation pairs, through which the information made in the whole camera networks is injected elegantly into the process of labeling inference.

Conceptually, the sampling space of labeling variables will grow unlimited, with the accumulation of observations, which may prevent the algorithm from being used in large-scale applications. However, it is notable that in most cases, the time separation between two successive observations of the same object cannot be arbitrarily large, and the number of objects is much smaller than that of observations. Accordingly, we set a limit of memory depth for each camera node by discarding the oldest observation, and control the size of sampling space of labeling variable by deleting elements with negligible posterior probability. In this way, we obtain an inference algorithm with constant computational and memory requirement, at very little cost of labeling accuracy.

Although many excellent works exist for object detection and tracking in single view, reliable detection in a crowded scene is still a challenging task. Sometimes missing detection may happen and the nonmissing detection assumption underlying our method is violated. We alleviate this problem by considering an enlarged neighborhood of camera, and assuming that object has been detected at least once in this neighborhood before it arrives to the current camera. We
use a mixture model to describe the uncertainty in object moving path caused by observation missing and modify the evaluation of spatiotemporal likelihood, leading to improved robustness of the algorithm against missing detection.

Extensive experiments are conducted on two datasets collected by our nonoverlapping camera networks: the office building dataset and campus garden dataset. Comparisons are made with two closed related inference-based consistent labeling algorithms. The results demonstrate that: (1) compared with centralized inference algorithm [1], our method shows significant superiority in execution speed, and achieves comparable labeling accuracy; (2) compared with distributed inference algorithm [2], our method provides the ability to estimate the number of moving objects and also shows obvious improvement in labeling accuracy; and (3) by considering the higher order neighborhood, our method gives satisfactory results in case of missing detections.

2. Problem Formulation

Suppose that multiple objects are moving in a large area monitored by $N$ smart cameras with nonoverlapping field of views, as shown in Figure 1. Here, the number of moving objects is not fixed in time and is unknown to us. We assume that each camera node has limited resource for computation, storage, and communication, and synchronized internal clocks that allow the nodes to share a common notion of time. The camera networks can be represented as a graph $G = (V, E)$, which is usually called camera networks topology or activity topology, see Figure 1. Each camera corresponds to a node $v \in V$ in the graph, and two nodes are linked by an edge $e \in E$ if the object can move from one node to another without being observed by other cameras. In this paper, camera node $v$ is called (zero-order) neighbor of node $u$ if $(u, v) \in E$. And we use $\mathcal{N}_v$ to denote the set of neighbors of $u$. The topological assumption imposes strong constraints on the movements of objects in the networks, in that any object, if not newly appeared, must be presented in the FOV of one of the current camera's neighbors before it arrives to the current camera. It holds true in many scenarios of interest and has been adopted in most of the works about tracking in nonoverlapping camera networks [1, 2, 10–17].

A clip of video is collected by camera when an object is passing through its FOV. We assume that the collected video clip has been summarized into a single virtual observation $y^k_{u,i}$, by some visual tracking and feature extraction algorithms running on the camera, such as [32, 33]. For convenience of discussion, here, the observation $y^k_{u,i}$ is double indexed: the local index $i$ implies that it is the $i$th observation generated by camera $u$, and the global index $k$ implies that it is the $k$th observation generated in the whole networks. It should be noticed that each camera is aware of the local index of the observations made on itself, but not the global index. Each observation $y^k_{u,i}$ consists of two parts: the appearance measurement $a^k_{u,i}$, such as the size, texture, color distribution, or biometric features of the object, and the spatiotemporal measurement $d^k_{u,i}$, such as the capturing time or the moving direction of the object in the camera's FOV. An example is shown in Figure 2. After a certain period of time, a set of observations are generated on each camera in the networks. We use $Y^{1:k}$ to denote the observation set generated in the whole networks up to step $k$.

For each observation $y^k_{u,i}$, we assign a random labeling variable $x^k_{u,i}$ to indicate which object the observation has originated from. The sampling space of $x^k_{u,i}$ consists of the identifiers, or labels (in this paper, the terms identifier and label are used interchangeably), of all possible objects that may produce observation $y^k_{u,i}$. And the labeling on different cameras should be consistent with each other; that is, observations generated by a single object on different cameras should be assigned with the same label. Our goal is to infer the posterior distribution of each labeling variables $x^k_{u,i}$, based on observations made in the whole networks up to the current time step, that is, $p(x^k_{u,i} \mid Y^{1:k})$. The label of each observation is determined by choosing the label value with the maximum posterior probability. And the camera-to-camera trajectory of each object can be recovered by associating the observations with the same label. Here, the main difficulties are (1) we cannot transmit all observations $Y^{1:k}$ to a central server for calculating $p(x^k_{u,i} \mid Y^{1:k})$, instead, the calculation must be based on local processing on individual cameras and communication between neighboring ones and (2) as the number of moving objects is unknown a priori, the sampling space of labeling variables on each camera should be determined and maintained consistently with each other on the fly and also in a distributed manner.

3. Inference Algorithm

In this section, we present our distributed Bayesian inference framework for consistent labeling. We show how to determine the sampling space of each labeling variable in Section 3.1 and how to perform inference in that space in Section 3.2. We present a distributed online algorithm with constant computation and memory requirements in Section 3.3 by limiting the memory depth of camera and maximum number of objects. Finally, we discuss the problem of missing detection and alleviate it by enlarging the camera's neighborhood.

3.1. The Sampling Space. We denote the sampling space of labeling variable $x$ as $\Gamma(x)$. The elements in $\Gamma(x)$ are globally unique identifiers of all possible objects generating the corresponding observation $y$. We assume that each observation $y$ is generated by single object on a specific camera. If multiple objects present in camera's FOV simultaneously, we can track them with multiobject tracking algorithm and produce one observation for each object. Thus, the pair $(u, i)$, where $u$ is the camera index and $i$ is the local time index, can be used as globally unique identifiers of objects. If an observation $y$ is labeled as $(u, i)$, this means that $y$ is originated from an object whose first observation in the networks is $y^k_{u,i}$. In other words, we use the head of the trajectory of the object as its identifier. As the pair $(u, i)$ is in one-to-one correspondence to the global
time index $k$; in the following discussion, we use $k$ in place of $(u, i)$ as element of $\Gamma$ just for simplicity in expression.

Suppose that at time step $k$, an observation $y_{u,i}^{k}$ is generated on camera $u$, and the corresponding labeling variable is $x_{u,i}^{k}$. Before step $k$, a set of observations has been made on $u$ and $u$’s neighbors. We denote this set as $Y_{s_u}^{k-1}$ and the corresponding labeling variable set as $X_{s_u}^{k-1}$. If no missing detection occurs, the observation $y_{u,i}^{k}$ either originates from object that has been detected in $u$’s neighborhood, or from a newly appeared one. Consequently, the sampling space of $x_{u,i}^{k}$ should be $k$ plus, the union of the sampling space of labeling variables in the set $X_{s_u}^{k-1}$; that is

$$\Gamma(x_{u,i}^{k}) = \{k\} \cup \left( \bigcup_{j \in I} \Gamma(X_{s_j}^{k-1}) \right),$$ (1)

where $X_{s_j}^{k-1}$ represents the $l$th element in the set $X_{s_j}^{k-1}$. We can see that by using (1), the sampling space of $x_{u,i}^{k}$ can be determined automatically based on local information on $u$ and its neighborhood. If no missing detection occurs, it can be ensured that each camera node is always performing inference in a sampling space consisting of identifiers of all possible objects that may generate the current observation.

3.2. The Posterior. In this subsection, we will discuss how to calculate the belief state; that is, the posterior distribution of the labeling variable over its sampling space, conditioned on all observations made in the whole networks up to the current time step. To clarify the interobservation dependence, for each $x_{u,i}^{k}$, we introduce a pointer variable $z_{u,i}^{k}$ to indicate the immediate predecessor of observation $y_{u,i}^{k}$ in the trajectory of a single object. It is easy to see that the sampling space of $z_{u,i}^{k}$ is $\{0, 1, \ldots, L\}$, where $L$ is the cardinality of the set $Y_{s_u}^{k-1}$. $z_{u,i}^{k} = 0$ means that $y_{u,i}^{k}$ is the first observation of a newly appeared object. $z_{u,i}^{k} = l \neq 0$ means that the last observation of the same object directly before $y_{u,i}^{k}$ is the $l$th element in the set $Y_{s_u}^{k-1}$. In this paper, we assume that the elements in $Y_{s_u}^{k-1}$ and $X_{s_u}^{k-1}$ have been sorted in reversed time order.

Using the Bayes rule, the joint belief state of $x_{u,i}^{k}$ and $z_{u,i}^{k}$ can be written as

$$b(x_{u,i}^{k} = h, z_{u,i}^{k} = l) \propto p(x_{u,i}^{k} = h, z_{u,i}^{k} = l) \times p(y_{u,i}^{k} \mid x_{u,i}^{k} = h, z_{u,i}^{k} = l, Y^{1:k-1})$$

$$= \lambda_{ap} \left( o_{u,i}^{k}, o_{u,i'}^{k'} \right) \lambda_{a} \left( d_{u,i}^{k}, d_{u,i'}^{k'} \right) p_{r} \left( x_{u,i}^{k} = h, z_{u,i}^{k} = l \right),$$ (2)

here, we assume that the $l$th observation in $Y_{s_u}^{k-1}$ is $y_{u,i'}^{k}$, and the appearance and spatiotemporal parts of the observation are conditionally independent given the hidden states. The term $\lambda_{ap}$ is the likelihood of appearance observation given that $z_{u,i}^{k} = l$,

$$\lambda_{ap} \left( o_{u,i}^{k}, o_{u,i'}^{k'} \right) \equiv p \left( o_{u,i}^{k} \mid z_{u,i}^{k} = l, Y^{1:k-1} \right),$$ (3)

it is a similarity measure between the pair of appearance observations $o_{u,i}^{k}$ and $o_{u,i'}^{k}$. In this paper, we take the normalized histograms of RGB brightness values within the object region as appearance observation and use the bidirectional Cumulative Brightness Transfer Functions (CBTF) proposed in [3] for calculating the appearance likelihood. We refer the readers to [3] for details of CBTF. By establishing a mapping of brightness values between camera views, the CBTF compensates the variation in illumination conditions at different camera sites. Here, we use CBTF for evaluating $\lambda_{ap}$ for its simplicity and effectiveness. In fact, any approach for the problem of appearance-based people reidentification across disjoint cameras [3–8], which is an interesting topic attracting intensive research efforts recently, can serve for this purpose as long as they can output a similarity measure between a pair of appearance observations.

The term $\lambda_{a}$ is the spatiotemporal likelihood given $z_{u,i}^{k} = l$. Supposing that the object entered the FOV of camera $u$ at time $t_{u,i}^{en}$ via frame border $e_{u,i}^{le}$ and left it at time $t_{u,i}^{en}$ via border $e_{u,i}^{le}$. The spatiotemporal likelihood can be written as

$$\lambda_{a} \left( d_{u,i}^{k}, d_{u,i'}^{k'} \right) \equiv p \left( d_{u,i}^{k} \mid z_{u,i}^{k} = l, Y^{1:k-1} \right)$$

$$= p \left( t_{u,i}^{en} \mid t_{u,i'}^{en} \right) p \left( e_{u,i}^{le} \mid e_{u,i'}^{le} \right).$$ (4)

Here, $d_{u,i}^{k} = (t_{u,i}^{en}, e_{u,i}^{en}, t_{u,i}^{le}, e_{u,i}^{le})$. The first factor $p(t_{u,i}^{en} \mid t_{u,i'}^{en})$ in (4) models the distribution of traveling time between camera $u$ and $u'$ as follows:

$$p \left( t_{u,i}^{en} \mid t_{u,i'}^{en} \right) = \begin{cases} 0, & \text{if } t_{u,i}^{en} < t_{u,i'}^{en} \left( + \Delta_{u,u'} \right), \\ N \left( (t_{u,i}^{en} - t_{u,i'}^{en}) \mid \delta_{u,u'}, R_{u,u'} \right), & \text{otherwise}, \end{cases}$$ (5)

where $N$ is Gaussian distribution, $\Delta_{u,u'}$ is the minimum travel time between $u$ and $u'$, $\delta_{u,u'}$ and $R_{u,u'}$ are the expected travel time and the variance of this distribution. Here, a truncated Gaussian distribution is used to prevent unrealistic travel time between cameras. The second factor $p(e_{u,i}^{en} \mid e_{u,i'}^{le})$ in (4) is a discrete distribution specifying the probability of an object arriving at camera $u$ via border $e_{u,i}^{le}$ when departing from camera $u'$ via border $e_{u,i'}^{le}$. The entries of this distribution and the parameters in (5), $(\Delta_{u,u'}, \delta_{u,u'}, R_{u,u'})$, are specified according to the prior knowledge of the layout of the region monitored by the camera networks, or learned from training data. When $z_{u,i}^{k} = 0$, there is no observation
before \( y_{u,i}^k \) in the same trajectory, the likelihood in (2) takes a constant value of \( \lambda_0 \), which is determined experimentally.

The prior in (2) can be factorized as follows:

\[
p_r (x_{u,i}^k = h, z_{u,i}^k = l) = p_r (x_{u,i}^k = h) p_r (z_{u,i}^k = l | x_{u,i}^k = h).
\]  

(6)

The term \( p_r (x_{u,i}^k = h) \) is the predictive distribution of \( x_{u,i}^k \) before observation \( y_{u,i}^k \) is available. From the nonmissing detection assumption mentioned before, \( x_{u,i}^k \) should either take value of \( k \) if \( y_{u,i}^k \) is from a newly appeared object, or take the same value as one element in \( X_{s}^{k-1} \) if \( y_{u,i}^k \) is from an older object. In the latter case, to predict the label of current observation, we first pick a variable \( X_{s}^{k-1} \), randomly from the set \( X_{s}^{k-1} \); then draw a labeling value according to its belief state, \( p(X_s^{k-1} | Y^{1:k-1}) \), which has been calculated in previous inference step. Consequently, the prior \( p_r (x_{u,i}^k = h) \) can be calculated as

\[
p_r (x_{u,i}^k = h) = \begin{cases} 
\frac{1}{L+1}, & h = k, \\
\frac{1}{L+1} \sum_{l=1}^{L} p(X_s^{k-1} = h | Y^{1:k-1}), & h \in \bigcup_{l=1}^{L} \Gamma (X_s^{k-1}).
\end{cases}
\]  

(7)

The term \( p_r (x_{u,i}^k = l | x_{u,i}^k = h) \) is the predictive distribution of pointer variable \( z_{u,i}^k \) knowing that the label of current observation is \( h \). From the definition of \( z_{u,i}^k \), it is easy to see that given \( X_s^{k-1}, z_{u,i}^k \) is independent of the information made on cameras beyond \( u \)’s neighborhood. Specifically, the prior \( p_r (z_{u,i}^k = l | x_{u,i}^k = h) \) can be written as

\[
p_r (z_{u,i}^k = l | x_{u,i}^k = h) = \begin{cases} 
p(X_s^{k-1} \neq h | Y^{1:k-1}), & l = 0, \\
p(X_s^{k-1} \neq h, X_s^{k-1} = h | Y^{1:k-1}), & l = 1, \ldots, L.
\end{cases}
\]  

(8)

In (8), we need to evaluate the joint distribution of elements in the set \( X_s^{k-1} \), which is intractable when \( L \) is large. Thus, we approximate it using the product of marginal distributions of individual labeling variables as follows:

\[
p(X_s^{k-1} | Y^{1:k-1}) \approx \prod_l p(X_s^{k-1} | Y^{1:k-1}).
\]  

(9)

The approximation scheme in (9) is similar with that of the Assumed Density Filter used in [1, 10], in that the joint belief of hidden variables is repeatedly factorized into an assumed density family, that is, the product of marginals, in each inference step. It is clear in the above discussion that the likelihood and prior can be calculated based solely on local information, that is, the observation made on camera \( u \) and its neighbors and the beliefs of the corresponding labeling variables.

By summarizing out the auxiliary variable \( z_{u,i}^k \) from (2), we get the belief of labels as follows:

\[
b(x_{u,i}^k = h) \propto p(x_{u,i}^k = h | Y^{1:k}) \propto p_r (x_{u,i}^k = h) \\
\times \sum_{l=0}^{L} \lambda_{ap} (o_{u,i}^k, o_{u,i}^{k', j'}) \lambda_{p} (d_{u,i}^k, d_{u,i}^{k', j'}) p_r (z_{u,i}^k = l | x_{u,i}^k = h).
\]  

(10)

In (10), the likelihood \( \lambda(y_{u,i}^k, y_{u,i}^{k', j'}) \) is a similarity measure between the current and previous observations. It can be viewed as a \((L + 1) - D \) vector, each component of which corresponds to a specific value of \( l \). Given \( x_{u,i}^k \) is available, the prior distribution of \( z_{u,i}^k \) can also be viewed as a \((L + 1) - D \) vector. It is interesting to note that the likelihood \( p(y_{u,i}^k | x_{u,i}^k) \) is the inner product of the above two vectors, which is proportional to the cosine of the angle between the two vectors, because the length of the vector \( \lambda \) is constant with respect to \( h \).

By calculating this angular metric [34], the observation similarity measure \( \lambda \) is transferred into likelihood scores for different label values; that is, \( p(y_{u,i}^k | x_{u,i}^k = h) \).

3.3. Limiting the Computational Cost. From (1), we can see that the sampling space of labeling variable increases linearly with inference step. At step \( k \), updating the belief state over sampling space costs \( o(L) \) for likelihood evaluation, \( o(kL) \) for prior and \( o(k) \) for posterior. Here, \( L \leq k \) is the number of observations made on \( u \)’s neighbors up to step \( k \), which is also increasing with time. The computational cost may become a limitation of the algorithm in large-scale application. We overcome this problem by setting a limit \( M \) of camera memory depth and a maximum size \( H \) of the sampling space.

As older observations are less likely to be the immediate predecessor of the current one, for inference in step \( k \), camera \( u \) only collects the most recent observations and the corresponding belief states from its neighbors. That is, we set a memory depth of \( M \) for each camera. Typically, the number of objects is much less than the number of observations made in the network. Thus, limiting the size of sampling space by an appropriately chosen \( H \) will not seriously affect the inference performance. At step \( k \), we prune the sampling space \( \Gamma(x_{u,i}^k) \) when its size exceeds \( H \) by deleting the element with the lowest posterior probability. In this way, we obtain a distributed inference algorithm with constant computational and memory requirements. The complete algorithm is summarized in Algorithm 1. We can see in Algorithm 1 that the inference is driven by the event of object detection. The algorithm advances one step further by inferring the label of the current observation. Moreover,
the inference involves only local information processing on each camera node and communication between neighboring cameras. Thus, it is a complete distributed algorithm.

3.4. Missing Detection. In the above discussion, we assume that objects can be detected reliably by smart cameras. In practice, however, false alarm and missing detection are always encountered due to unfavorable observing conditions. For consistent labeling, the problem of false alarm can be dealt with simply by deleting observations with likelihood below a specified threshold. On the other hand, the problem of missing detection is more critical and difficult to treat, since in this case the neighboring structure of the camera networks topology is destroyed, and the assumption underlying our distributed inference is violated. Thus, for the sake of brevity, we focus on missing detection only in this paper.

We partially overcome the problem of missing detection by considering information on the enlarged neighborhood of camera \( u \) when the label of observation made on \( u \) is inferred. We denote the enlarged neighborhood as \( N_{u}^{q} \), and call it as \( q \)-order neighborhood of \( u \). \( N_{u}^{q} \) consists of all cameras \( v \) in the networks where there exists at least one path between \( u \) and \( v \) with length no longer than \( q \). The path length \( q \) is defined as the number of camera nodes along the path between \( u \) and \( v \), and \( q = 0 \) means that \( u \) and \( v \) are connected directly. We assume that object cannot reach camera \( u \) without being detected by cameras in \( N_{u}^{q} \) if it is not a newly appeared one. In this case, the spatiotemporal likelihood should be evaluated by the following mixture model:

\[
\lambda_{st}^{q}(d_{st}^{k}, \hat{d}_{st}^{k'}) = \omega_{0}^{0} \lambda_{st}^{0}(d_{st}^{k}, \hat{d}_{st}^{k'}) + \sum_{r=1}^{q} \omega_{r}^{0} \lambda_{st}^{0}(d_{st}^{r}, \hat{d}_{st}^{r'}),
\]

(11)

the weight parameters \( \omega \) correspond to different paths chosen by object when it is moving from \( u' \) to \( u \) : \( \omega_{0}^{u'u} \) which is the probability of object moving along the edge \((u', u)\), which is the transition probability defined on the network topology. If \((u', u)\) does not exist, \( \omega_{0}^{u'u} = 0 \), \( \omega_{r}^{u'u} \) is the probability of object moving along the \( j \)th path from \( u' \) to \( u \) with length \( r \), which is the normalized product of the transition probabilities of the \( r + 1 \) edges along that path. \( \lambda_{st}^{0} \) is defined by (4), and \( \lambda_{st}^{q} \) is the higher-order spatiotemporal observation model, which can be determined from the knowledge of monitored area, or directly from zero-order models. For example, if the traveling time along edges follows Gaussian, the traveling time along a \( r \)-length path also follows Gaussian, with mean and variance equaling to the sum of parameters corresponding to the \( r + 1 \) edges in that path.

To apply the inference algorithm to the case of missing detection, we only need to replace the zero-order neighbors \( N_{u} \) with higher-order neighbors \( N_{u}^{q} \) in Algorithm 1, and evaluate the spatiotemporal likelihood by (11) instead of (4). After these modifications, the posterior of the labeling variable is inferred based on more information than before, leading to improved robustness against missing detection at the cost of increased communication and computation at each camera node.

4. Results

4.1. Experiment Settings. In this section, we report the experimental results of the proposed algorithm in two different disjoint multicamera surveillance scenarios. Object detection within a single view is based on the background subtraction and shadow removal algorithm as proposed in [32]. Object tracking in each camera is based on the probabilistic, appearance-based tracking algorithm as proposed in [33]. When a person is passing through the FOV of a camera, an observation is extracted from the collected video as described in Section 2.

Scenario 1: Office Building Experiment. The experiment in scenario 1 was conducted with ten cameras mounted in an office building, five in the first floor and five in the second floor. The camera layout and the corresponding topology are shown in Figures 3(a) and 3(b). A total of 300 observations originated from 10 persons are extracted from the 70-minutes video data collected by the cameras in the networks. The main challenges are the significant variations in illumination and view angles. For example, the areas covered by camera B, E, and I are clearly dim due to the lack of lighting. And the view...
angles at stairs, C and F, are quite different from those at other camera sites.

Scenario 2: Campus Garden Experiment. The camera networks used in this experiment consist of ten cameras mounted in our campus garden, and the layout and corresponding topology are shown in Figures 3(c) and 3(d). During the experiment, 14 persons are present in the monitored region. We gather altogether 300 observations of them from 90-minutes video collected by the camera networks. In outdoor scenario, the illuminating conditions at each site are similar, but the monitored region is larger and the distance between cameras is longer than that of indoor case. Consequently, the variation of traveling time is large and the discriminating power of traveling time cue is decreased.

Before conducting experiments, the observations extracted from about five hours-long video data collected by the camera networks are manually labeled and used for learning the CBTF and traveling time model \( p(e_{	ext{en}} | t_{	ext{le}}) \) across neighboring cameras. The moving direction transition probabilities \( p(e_{	ext{en}} | e_{	ext{le}}) \) are specified according to the layout of cameras in the monitored region. Figure 4 shows the learned CBTF in RGB channels between cameras A and B in office building experiment. The CBTF was learned using 79 training pairs. Note that as camera site B is obviously dimmer than camera A, mostly lower color values from camera B are being mapped by the CBTF to higher color values in camera A. Figure 5 shows the normalized traveling time histogram between camera A and C in campus garden experiment, and the fitted Gaussian distribution. A total of 82 transitions between the two camera sites were recorded. Note that the traveling time is almost between 40 and 80 seconds. We assume that human could not walk arbitrarily fast. So, we set a truncating threshold on the left side of Gaussian distribution. To make it safe enough, we set the threshold to one half of the minimum traveling time in the training data between each pair of neighboring camera sites.

4.2. Evaluation Criteria. We use the following measures to evaluate the algorithms: the estimated number of objects \( K \), the precision \( P \), recall \( R \), and \( F \)-measure of the reconstructed trajectories. Let \( \{Y_1^*, \ldots, Y_K^*\} \) be the ground truth, each of which is a set of observations that belong to the same object. Let \( \{Y_1, \ldots, Y_K\} \) denote the mutually disjoint subsets of \( Y \).
each of which is a set of observations assigned with the same label. The precision and recall of a set of estimated trajectories is defined as

\[
P = \frac{1}{K} \sum_{i=1}^{K} \max_j \left\{ \frac{|Y_i \cap Y_j^*|}{|Y_i|} \right\},
\]

\[
R = \frac{1}{K} \sum_{i=1}^{K} \max_j \left\{ \frac{|Y_i \cap Y_j^*|}{|Y_j^*|} \right\}, \quad F = \frac{2 \cdot P \cdot R}{P + R}.
\]

The precision and recall represent the fidelity and the completeness of the estimated trajectories, respectively. A reconstructed trajectory with a single observation has a 100% precision, and a reconstructed trajectory with all observations has a 100% recall. And the F-measure is the harmonic mean of these two measures.

To evaluate the speed of the labeling algorithm, we measure its execution time \( \tau(Y) \) on the observation set \( Y \). The time cost involved with data collection, communication, object segmentation, and tracking of object on single camera are ignored in this paper. For centralized algorithm, we use \( \tau_c(Y) \), the execution time when the algorithm runs on a single machine, as speed measure. For distributed algorithm, which allows each camera agent to independently process data, we use \( \tau_d(Y) \), the maximum of the execution time of individual agent, as speed measure.

4.3. Results. We apply our method to the two datasets. In Office Building experiment, we set the memory depth as \( M = 20 \) and the sampling space limit as \( H = 15 \). In Campus Garden experiment, we set them as \( M = 25 \) and \( H = 20 \), respectively. The inference results are shown in...
Figure 6: Marginal distribution of labeling variable. Each column corresponds to one observation, sorted in time order. The true label of each observation is depicted by red star. Grayscale corresponds to the posterior probability of labeling variables. Black represents probability 1, and white 0. (a) Results on office building dataset and (b) results on campus garden dataset.

Table 1: Results on office building data ($K^* = 10$).

| $\tau_c$ (s) | $\tau_d$ (s) | $K$ | Precision (%) | Recall (%) | $F$ measure (%) |
|-------------|--------------|-----|---------------|------------|-----------------|
| Zajdel [1]  | $2.9 \times 10^5$ | X   | 11            | 86.78      | 84.66           | 85.71           |
| Jiuqing and Qingyun [2] | X | 10.89 | 13            | 75.96      | 69.00           | 72.31           |
| Ours        | X            | 5.82 | 10            | 92.84      | 92.00           | 92.42           |

$K^*$: represents the ground truth.

Comparison with Centralized Inference Algorithm [1]. A closely related method to ours was proposed in [1], in which the joint distribution of observations and hidden variables is encoded by dynamic Bayesian networks, and the posterior marginal of labeling and counting variables are inferred by using the Assumed Density Filter. By inferring the counting variables, the number of objects can be estimated from observations automatically. In [1], the appearance of a single
Table 2: Results on campus garden data ($K^* = 14$).

|            | $\tau_c$ (s) | $\tau_d$ (s) | $K$ | Precision (%) | Recall (%) | $F$ measure (%) |
|------------|--------------|--------------|-----|---------------|------------|-----------------|
| Zajdel [1] | $3.0 \times 10^3$ | X            | 16  | 95.44         | 97.62      | 96.52           |
| Jiuqing and Qingyun [2] | X          | 12.00        | 19  | 80.94         | 89.95      | 85.65           |
| Ours       | X            | 5.80         | 14  | 90.54         | 93.57      | 92.03           |

$K^*$: represents the ground truth.

Figure 7: Selected frames in Office Building experiment. Column corresponds to camera site and row corresponds to time instant. Detected person is shown with bounding box, the label of which is shown in the text box. Left-top, true label; right-top, result of Zajdel [1]; left-bottom, result of Jiuqing and Qingyun [2]; right-bottom, our result.

A person under different camera sites is assumed to follow a single Gaussian with Normal-Inverse Wishart distributed mean and covariance parameters, allowing the appearance model to be updated analytically. However, the inference algorithm in [1] is centralized, and the computational and memory cost increase rapidly with number of observations. To make it executable on our datasets, we limit the memory depth of the central server to 25 for both experiments, and set the maximum number of objects as 15 and 20 for the two experiments, respectively. It is obvious in Tables 1 and 2 that...
the speed of [1] is much slower than ours, mainly due to its centralized nature. It is also noticeable that the centralized method [1] does not show superiority to our distributed method on the Office Building data in terms of labeling accuracy, due to the unrealistic appearance assumption and truncating of history observations during inference. In fact, for centralized inference, a much deeper memory is required to ensure that the true predecessor of the current observation is reserved. But this will lead to unacceptable computational and memory cost.

Comparison with Enhanced Distributed EM [2]. We also compare our method with [2], in which a distributed inference algorithm based on the same nonmissing detection assumption as ours is proposed to calculate the posterior distribution of labeling variables conditioned on both appearance and spatiotemporal observations. The appearance of a single person is assumed to follow a single Gaussian, which is updated in the M-step of distributed EM framework. The method in [2] is offline, and requires the number of objects to be prefixed. In our experiments, we set the maximum object number as 15 and 20, respectively, run EM 30 iterations, and initialize the appearance model by $k$-means clustering. As shown in Tables 1 and 2, [2] cannot estimate the number of persons correctly in both experiments, and gives significantly lower labeling accuracy than our method. The inferior performance of [2] can be attributed to its single
Gaussian assumption on appearance and its blindness in choosing the sampling space of labels. In experiments, we observe that by using [2], the trajectory of a single person tends to break into several pieces especially when his/her appearance undergoes obvious changes across camera sites, and consequently more trajectories are recovered than the ground truth. In contrast, our method only assumes that the appearance of a single person at each pair of neighboring cameras is “similar” to each other after the CBTF transform. By using CBTF, the variations in lighting conditions across different camera sites can be partially compensated. This assumption seems more reasonable than that in [2], and results in an improved labeling accuracy. For example, in Figure 7, person e presented consecutively in row 6, column 1, row 8, column D, and row 12, column H. The method in [2] cannot correctly label the person at camera I mainly due to the weak lighting condition. However, our method can give correct results in this case. Of course, if the observing angle or the attitude of person varies significantly, simple appearance features such as brightness histogram used in this paper are not enough. In this case, more robust appearance features and advanced similarity evaluation techniques (such as in [3–8]) are required. It is noticeable that our framework is rather flexible in that these new similarity evaluation techniques can be easily incorporated into the inference framework by modifying the likelihood function.

4.4. Missing Detection. In our experiments, no missing detection occurs as the scenes are relatively sparse. However, in practice, the interested objects may be missdetected due to occlusions in crowd or the low quality of the video. To verify our method in these cases, we randomly delete some of the observations, and apply the proposed algorithm (0-order spatiotemporal model (4)) and its modification (1-order spatiotemporal model (11)), respectively, to the remaining part of the two datasets. The average $F$-measure of 10 trials is shown in Figure 9. As expected, when missing detection occurs, the labeling accuracy of 0-order model-based algorithm decreases rapidly. When 40 observations are deleted randomly (corresponding to a 13% missing rate), 0-order method gives $F$-measures of 59% and 62% for the two datasets. In contrast, by enlarging the neighborhood, 1-order method can achieve $F$-measures of 77% and 82% in this case. This can be attributed to the fact that in case of missing detection, the real predecessor of current observation is more likely to exist in higher-order neighborhood than in 0-order neighborhood; hence, a correct link is more likely to be established by considering higher order neighborhood.

4.5. Discussion. In our experiments, the scenes are rather sparse, so that the persons can be easily segmented and tracked within a single view by standard multiobject tracking algorithms. When the scene becomes crowded, more occlusions will occur. This leads to the following two problems. First, occlusions may lead to missing detections. We have addressed this problem in last subsection by considering higher-order neighborhood and using mixture spatiotemporal model. Second, in case of heavy occlusion, it is more difficult to segment and extract accurate features of persons. In this case, the performance of our method will deteriorate. This is a limitation of our method. In the future, we will try to use more advanced pedestrian detection technique and more reliable feature extraction algorithms to deal with crowded scenarios.

In our method, we use both appearance and spatiotemporal cues for inference, which are complementary to each other. However, the use of traveling time cue is an option of both benefits and downside. If the variation in walking speed of persons under tracking is not very large, using traveling time cue can improve the labeling performance. On the other hand, if some person walks extremely fast or slow, using traveling time evidence may deteriorate the performance. In our experiments, there is no running action, but there is some short pause between camera sites, which makes the traveling time cue less reliable. In these cases, the inference relies mostly on the appearance cue, and another spatiotemporal cue, that is, the moving direction of persons, which impose strong constraints on person’s trajectory. In our experiments, we find that in most cases, our method can label observation correctly even when some short staying occurs.

However, in practice, person may move arbitrarily slow, for example, hiding in the unmonitored region for a long time. In fact, if a person has a long-time staying when he/she traveled from one camera to another, the traveling time likelihood evaluated by (5) is approaching zero. In this case, the labeling result is very likely to be incorrect. To cope with this problem, we consider the following traveling time model:

$$p\left(t^e_{ij} \mid t^l_{ij} \right) = \begin{cases} 0, & \text{if } t^e_{ij} - t^l_{ij} \leq \Delta_{ij}, \\ 1, & \text{otherwise}, \end{cases}$$

that is, we only impose a lower bound constraint on the traveling time. This traveling time constraint is much weaker than that in (5), but it is more suitable when person walks in abnormal speed, for example, stays for a long time during walking. By using this traveling time model, it is possible to
To verify this point, we simulate the random stays during walking by adding values drawn uniformly from time interval 30–600 s to the traveling times data in the original datasets, and apply our method to the resulting data. We present the simulation results in the following table, in which the $F$-measure is the average of 10 trials.

In Table 3, “app + Gaussian + direction” means that we use the appearance, moving direction and traveling time models; that is, (3), (4), and (5) for inference, “app” means that we use only appearance model, and “app + Uniform + direction” means that we replace the Gaussian traveling time model (5) with the uniform model (13). It can be seen from the table that in case of possible long-time staying, the labeling results based on Gaussian traveling time model are very poor, mainly due to the unrealistic traveling time assumption. And labeling based only on appearance information is also unsatisfactory. In contrast, by using the uniform traveling time model, spatiotemporal information can be used in an effective way, and significant improvement in labeling accuracy is achieved. This demonstrates the flexibility of our frameworks. We can choose proper models according to the situation without changing the algorithm.

4.6. Failure Case. In our experiments, most of the observations can be labeled correctly. This is mainly originated from the complementarity between the appearance and spatiotemporal evidences. However, if both of them are direct to wrong associations, or the two kinds of evidence contradict each other and the evidence leading to correct answer is not strong enough, incorrect labeling tends to occur. For example, as shown in Figure 10, observations 1 and 2 on camera A were originated from person e and c, respectively. And observation 3 on camera E was originated from person e. Actually, observation 1 is the true predecessor of observation 3. However, as persons b and d look quite similar, the appearance cue is somewhat misleading. As shown in the figure, the appearance similarity between observations 1 and 3 is 0.0042, and that between 2 and 3 is 0.0231. On the other hand, since observations 1 and 2 were generated on camera A very closely in time, the spatiotemporal evidence is less discriminating. Indeed, the spatiotemporal similarity between observations 1 and 3 is 0.0515, while that between 2 and 3 is 0.0230. Consequently, the overall similarity between observations 1 and 3 is less than that between 2 and 3, and we find that the observation 3 is mislabeled as c by our algorithm.

5. Conclusion

In this paper, we present a distributed Bayesian inference framework for consistent labeling of tracked objects in nonoverlapping camera networks. In this method, each camera in the networks performs inference on labeling variables over the online determined sampling space, based on local information and that collected from its neighbors. The similarity between pairs of observations is used for defining likelihood function in the framework, making it very flexible and particularly suitable in case of large observing condition variations across camera views. To cope with missing detection, we enlarge the neighborhood of each camera from which it collects information during inference and use a higher order mixture model to evaluate spatiotemporal likelihood, leading to improved robustness of the algorithm. The effectiveness of the proposed method is verified on two real datasets. In the future, we plan to investigate how to extend the use of our method to more realistic scenarios, in which the size of networks is larger, the duration of video collection is longer, and the camera scene is more crowded.

Acknowledgments

The authors are grateful to the student volunteers for their participation in the tracking experiments. This work is supported by the National Natural Science Foundation of China, under Grant no. 61174020. The authors would like to thank the anonymous reviewers for their valuable suggestions for improving the quality of the paper.

References

[1] W. Zajdel, Bayesian visual surveillance: from object detection to distributed cameras [Ph. D. dissertation], University of Amsterdam, Amsterdam, The Netherlands, 2006.

[2] W. Jiuqing and L. Qingyun, “Distributed data association in smart camera networks,” in Proceedings of the 5th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC'11), IEEE, August 2011, (An extended version of this paper, Distributed data association in smart camera networks using belief propagation, has been accepted by ACM
Transactions on Sensor Networks, to be appeared in May, 2014.

[B] B. Prosser, S. Gong, and T. Xiang, "Multi-camera matching using bi-directional cumulative brightness transfer functions," in Proceedings of the British Machine Vision Conference, vol. 463, BMVA Press, 2008.

[W. S. Zheng, G. Shaogang, and X. Tao, "Reidentification by relative distance comparison," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 3, pp. 653–668, 2013.

[S. Baik, G. Charpiat, E. Corvée, F. Brémond, and M. Thonnat, "Learning to match appearances by correlations in a covariance metric space," in Computer Vision—ECCV 2012, vol. 7574, pp. 806–820, Springer, Berlin, Germany, 2012.

[M. Hirzer, P. M. Roth, M. Kostinger, and H. Bischof, "Relaxed pairwise learned metric for person re-identification," in Computer Vision—ECCV 2012, pp. 780–793, Springer, Berlin, Germany, 2012.

[B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang, "Person re-identification by support vector ranking," in Proceedings of the British Machine Vision Conference, vol. 10, BMVA Press, 2010.

[R. Layne, T. Hospedales, and S. Gong, "Person re-identification by attributes," in Proceedings of the British Machine Vision Conference, vol. 2, BMVA Press, 2012.

[G. Lian, J. Lai, and W. Zheng, "Spatial-temporal consistent labeling of tracked pedestrians across non-overlapping camera views," Pattern Recognition, vol. 44, no. 5, pp. 1121–1136, 2011.

[T. Wan and Q. Liu, "Efficient data association in visual sensor networks with missing detection," Eurasip Journal on Advances in Signal Processing, vol. 2011, Article ID 176026, 42 pages, 2011.

[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.

[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.

[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), IEEE, September 2009.

[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), IEEE, September 2009.

[T. E. Choe, Z. Rasheed, G. Taylor, and N. Haering, "Globally optimal target tracking in real time using max-flow network," in Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV ’2011), pp. 1855–1862, IEEE, November 2011.

[D. M. Antunes, D. Figueira, D. M. Matos, A. Bernardino, and J. Gaspar, "Multiple Hypothesis Tracking in camera networks," in Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV ’11), pp. 367–374, IEEE, November 2011.

[B. C. Matei, H. S. Sawhney, and S. Samarasekera, "Vehicle tracking across nonoverlapping cameras using joint kinematic and appearance features," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’11), pp. 3465–3472, IEEE, June 2011.

[T. Mensink, W. Zajdel, and B. Kröse, "Distributed EM learning for appearance based multi-camera tracking," in Proceedings of the 1st ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC ’07), pp. 178–185, IEEE, September 2007.

[R. D. Nowak, "Distributed EM algorithms for density estimation and clustering in sensor networks," IEEE Transactions on Signal Processing, vol. 51, no. 8, pp. 2245–2253, 2003.

[A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," Proceedings of the IEEE, vol. 98, no. 11, pp. 1847–1864, 2010.

[D. Gu, "Distributed EM algorithm for Gaussian mixtures in sensor networks," IEEE Transactions on Neural Networks, vol. 19, no. 7, pp. 1154–1166, 2008.

[M. Çetin, L. Chen, J. W. Fisher III et al., "Distributed fusion in sensor networks," IEEE Signal Processing Magazine, vol. 23, no. 4, pp. 42–55, 2006.

[T. Anker, D. Doele, and B. Hod, "Belief propagation in wireless sensor networks—a practical approach," in Wireless Algorithms, Systems, and Applications, pp. 466–479, Springer, Berlin, Germany, 2008.

[A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky, "Non-parametric belief propagation for self-localization of sensor networks," IEEE Journal on Selected Areas in Communications, vol. 23, no. 4, pp. 809–819, 2005.

[H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," Proceedings of the IEEE, vol. 97, no. 2, pp. 427–450, 2009.

[J. Schiﬀ, D. Antonelli, A. G. Dimakis, D. Chu, and M. J. Wainwright, "Robust message-passing for statistical inference in sensor networks," in Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN ’07), pp. 109–118, IEEE, April 2007.

[M. Paskin, C. Guestrin, and J. McFadden, "A robust architecture for distributed inference in sensor networks," in Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN ’05), pp. 55–62, IEEE, April 2005.

[L. Chen, M. J. Wainwright, M. Çetin, and A. S. Willsky, "Data association based on optimization in graphical models with application to sensor networks," Mathematical and Computer Modelling, vol. 43, no. 9, pp. 1114–1135, 2006.

[S. Calderara, R. Cucchiara, and A. Prati, "Bayesian-competitive consistent labeling for people surveillance," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, pp. 354–360, 2008.

[A. Kamal, J. A. Farrell, and A. Roy-Chowdhury, "Information consensus for distributed multi-target tracking," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’13), 2013.

[M. Taj and A. Cavallaro, "Distributed and decentralized multi-camera tracking," IEEE Signal Processing Magazine, vol. 28, no. 3, pp. 46–58, 2011.

[R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, pp. 1337–1342, 2003.

[R. Cucchiara, C. Grana, G. Tardini, and R. Vezzani, "Probabilistic people tracking for occlusion handling," in Proceedings of the 17th International Conference on Pattern Recognition (ICPR ’04), vol. 1, pp. 132–135, IEEE, August 2004.
[34] S. H. Cha, “Comprehensive survey on distance/similarity measures between probability density functions,” International Journal of Mathematical Models and Methods in Applied Sciences, vol. 1, no. 4, pp. 300–307, 2007.
Submit your manuscripts at
http://www.hindawi.com