Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 591067, 15 pages
http://dx.doi.org/10.1155/2015/591067

Research Article

Multiple People Tracking Using Camera Networks with Overlapping Views

Wan Jiuqing and Li Achuan

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

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

Received 29 July 2014; Revised 23 December 2014; Accepted 31 December 2014

Academic Editor: Janez Perš

We present a novel framework for multiple pedestrian tracking using overlapping cameras in which the problems of object detection and data association are solved alternately. In each round of our algorithm, the people are detected by inference on a factor graph model at each time slice. The outputs of the inference, namely, the probabilistic occupancy maps, are used to define a cost network model. Data association is achieved by solving a min-cost flow problem on the resulting network model. The outputs of the data association, namely, the ground occupancy maps, are used to control the size of factors in graph model in the next round. By alternating between object detection and data association, a desirable compromise between complexity and accuracy is obtained. Experiments results on public datasets demonstrate the competitiveness of our method compared with other state-of-the-art approaches.

1. Introduction

In recent years, there are lots of research interests in multiple people detection and tracking using camera networks with overlapping views. Generally speaking, proposed methods can be categorized into image based ones and state based ones. In the image based methods [1–4], typically, some multiview geometry is utilized for people localization using the foreground detection results in multiple cameras. In [1, 2], the vertical axes of the foreground blobs corresponding to a single person in different views are warped into a common view using ground plane homography, and the ground position of the person is determined by the intersection of the mapped axes. Similarly, in [3] the foreground likelihood maps in different views are warped into a common reference view to generate a synergy map, in which the occupying positions are assigned with higher values. In [4], the vertical axis of a person is projected from one view to another by jointly using the ground plane homography and epipolar geometry. The main drawback of the image based methods is that the localization performance depends heavily on the foreground segmentation quality and deteriorates rapidly when complex occlusions occur. In contrast, occlusions can be treated naturally in state based methods [5–7], in which the ground plane is discretized into regular grid and the occupancy states on each position are estimated by inference on a MRF (Markov random field) or CRF (conditional random field) model. The main difficulties in state based methods are the combinatorial explosion of the state space and the higher-order dependency structure in the underlying graphical model. Therefore, the state based methods have to resort to either random sampling algorithms [5] or assumption of simplified structure in graphical models [6, 7]. In [6], the ground is discretized into regular locations and the occupancy state is represented by introducing a Boolean variable. With the hypothesis of independency between each of the locations, the joint posterior occupancy distribution is approximated by the product of marginal probability of the occupancy state on each location by minimizing the KL (Kullback-Leibler) divergence. Performance of the method significantly decreases in case of intense crowd due to the independent hypothesis. In [7], the dependency caused by occlusion between people is modeled by higher-order MRFs and the occupancy state is determined by cascaded optimization on a sequence of MRFs with increasing

clique size. The optimization problem in each stage, that is, in each MRF with specific clique size, is solved using the pattern based optimization algorithm. And the optimizer of the lower order MRF is used to impose constraint on the state space of the higher-order MRF in the next level, in order to maintain the computation to be tractable. Finally, by considering the whole ground plane as a single clique, the global optimizer can be obtained. However, when the monitored area is crowded with people, the speed of the algorithm [7] will be slowed down significantly. In fact, there are several works that deal with occlusions by inferring on a graphical model.

The tracking-by-detection method has the inherent weakness that it requires the output of the detectors to be reliable in order for the data association module to work properly. However, fast and reliable detection in crowded scene is still an unresolved problem. It is advantageous to solve for detections and data association jointly, rather than computing detections first and then linking them into trajectories. The method presented in [8] uses quadratic Boolean programming to solve detection and tracking in coupled manner. Multiple tracking hypotheses are kept and the most likely trajectories are found by searching forward as well as backwards in time. The authors in [9] expressed the problems of detection and tracking in optimizing a single objective function and solved the optimization problem using Lagrange dual decomposition strategy. Specifically, the subproblem of detection is solved by using spars recovery technique and subproblem of data association is solved by network flow and the two subproblems reach consensus iteratively by projected subgradient optimization.

Inspired by the joint detection and tracking strategy, in this paper we proposed a new method based on previous work [7] for multiple people tracking using camera networks with overlapping views. The main drawback of [7] is its speed, especially in case of crowded scenarios. The key to speed up [7] is to eliminate false alarms generated by inference on MRF with small-sized cliques as fast as possible. In this work we try to accelerate the elimination process in two ways. Firstly, we exploit spatiotemporal information to delete false alarms by running object detection and data association modules alternatively. But we note that the performance of data association relies heavily on the probabilistic distribution of occupancy state over the monitored area. So secondly, we incorporate the pedestrian detector into our factor graph model for occupancy map inference, resulting in a more peaky distribution. In fact, addition of pedestrian detector to foreground mask to infer the occupancy maps has been studied in several works. The main features of our work include the following.

(1) In each time slice, a probabilistic occupancy map is calculated by inference on a factor graph model using belief propagation algorithm [10]. The potential function of the factor graph is defined using output of a HOG-based person detector and a generative model that compares back-projected synthetic images with foreground masks. By inferring on factor graph model, information from multiple views is fully exploited, and the problem of occlusion between people is well addressed.

(2) In the data association phase, the probabilistic occupancy maps are used to define a cost network model, and the ground occupancy maps in each time slice are determined by solving a min-cost flow problem on the cost network model using the efficient k-shortest paths algorithm.

(3) Our algorithm works iteratively. The resulting ground occupancy maps in current round are used to determine the elements contained in each factor in the next round. In each round, each factor contains only state variables corresponding to 1-state (occupied) locations in the ground occupancy map within a fix-sized sliding window. Along with iteration, the number of occupied positions in ground occupancy maps decreases monotonously. Thus the factor window can be enlarged gradually, taking into account occlusions in larger and larger area, while maintaining the computation to be tractable. To avoid the computational complexity in inference, we begin with small factor window in the first round.

(4) The experimental results show that our algorithm provides a desirable compromise between complexity and accuracy. By alternating between object detection and data association, the size of the factors can be well controlled. Compared with the state-of-the-art methods, it can reduce false alarms and missed detections in some cases.

The rest of the paper is organized as follows. In Section 2 we formulate the problem of detecting people using multiple cameras as an inference problem on factor graph model and define the potential functions in the factor graph model. In Section 3 we present our framework in which person detection (calculating probabilistic occupancy map) and data association (calculating ground occupancy map) are performed alternatively and show how to solve the two problems using belief propagation and k-shortest path algorithms, respectively. Experiment results are reported in Section 4 and conclusion are given in Section 5.

2. Multipeople Localization

In this section we deal with the problem of multiple people localization using camera networks with overlapping views, as shown in Figure 1. In case of intensive crowd and frequent occlusions, multipeople localization and tracking using single camera are difficult to achieve ideal performance. However, the comprehensive utilization of multiple cameras can significantly improve the accuracy of both.

Assume that J cameras are simultaneously available. Here $O_t = \{O_t^1, \ldots, O_t^J\}$ stands for the observations at time t and O stands for the entire observations from $t = 1$ to $t = T$. As in previous works [6, 7, 11], we discretize the visible part of the ground plane into a finite number of regularly spaced 2D locations $\{1, \ldots, K\}$. For each location k at each time t we

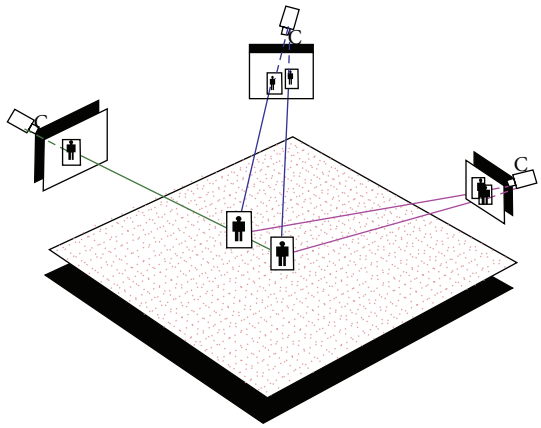


Figure 1: The scheme of multicamera localization.

introduce a Boolean variable m_t^k to indicate the occupancy state, and $m_t^k = 1$ means that there is a person on location k at time t. The localization problem can be formulated as calculation of the marginal posterior probability $\mu_t^k = p(m_t^k = 1 \mid O_t)$.

2.1. Factor Graph Model. Factor graph [10], a bipartite graph that expresses which variables are arguments of which local function, is a graphical model that can be used to visualize the factorization of a complicated probabilistic distribution. Assume that the joint posterior probability distribution of m_t can be factorized as follows (subscript t is omitted in this session for clarity):

$$p(M \mid O) \propto \prod_a f_a(M_a), \tag{1}$$

where $M = \{m^1, \ldots, m^K\}$ stands for the set of occupancy state variables corresponding to all locations on the ground and $M_a = \{m^l, l \in a\}$ stands for the set of occupancy state variables belonging to factor a. Here a is the index of factor.

The potential function $f_a(M_a)$ is defined as

$$f_a(M_a) = \exp\left(\omega_1 \times \frac{1}{L}\sum_{l=1}^{L}\frac{1}{J}\sum_{j=1}^{J}\varphi^j(m^l)\right) + \exp\left(\omega_2 \times \frac{1}{J}\sum_{j=1}^{J}\psi^j(M_a)\right), \tag{2}$$

where φ is a function of occupancy state variable m^l, predicting the probability of the presence of person at location l, and ψ is a function of the set of occupancy state variables M_a, modeling the joint image likelihood considering occlusions between pedestrians. ω_i $(i = 1, 2)$ are the weights of the two terms. In this paper, we assume that observations from different views are independent from each other.

2.2. Potential Function. In this subsection we define the potential function φ and ψ in (2). We model human shape using cylinders with fixed radius and height. From camera calibration, we obtain for each camera j a family of rectangular shapes by placing single cylinder on each possible location, as shown in Figure 2(a). Furthermore, the rectangular shape is normalized to the detection window I_l of size 64×128, as shown in Figure 2(b).

Potential function $\varphi^j(m^l)$ indicates the probability of the presence of pedestrian within the detection window I_l, as follows:

$$\varphi^j(m^l) = p(m^l = 1 \mid I_l). \tag{3}$$

The probability is given by pedestrian detector. Firstly, we choose Histograms of Oriented Gradients (HOG) as feature vector, which represents the outline information of pedestrian [12]. For each detection window, a 3780-dimensional feature vector is obtained. Then we use support vector machine (SVM) as classifier, train pedestrian model using public database, and finally achieve SVM decision function expressed as follows:

$$g(x) = \sum_{i=1}^{N_s} a_i y_i K(x_i, x) + b, \tag{4}$$

where x is the input feature vector, N_s is the number of support vectors, $a_i y_i$ is the coefficient of the ith support vector, b is the offset, and $K(x_i, x)$ is the kernel function. Based on the output of (4), the presence probability is determined according to the method proposed in [13].

Occupancy state on all locations determined simply by pedestrian detector is difficult to achieve ideal effect due to the occlusions between pedestrians. So we must consider the interaction of occupancy states between different locations, which is caused by occlusions in certain view. Next we show the generation model of observation B^j of a single view given the occupancy state of factor M_a. We obtain the synthetic image A_l^j for each camera j at each location l, as shown in Figure 3(c). Given the realization of M_a, a synthetic foreground mask can be produced by putting rectangles at locations where $m^l = 1$. The synthetic image $A^j(M_a)$ can be written as

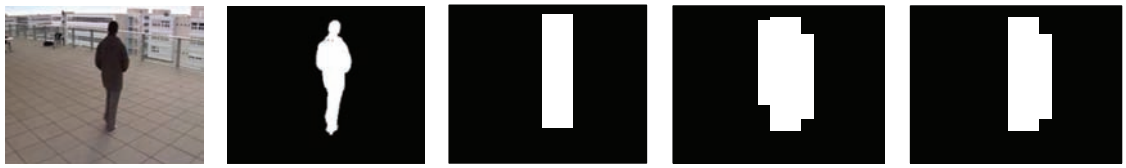
$$A^j = \bigoplus_l m^l A_l^j, \tag{5}$$

where $\oplus$ denotes a union operator. Note that the synthetic image A^j is the function of M_a. The observation B^j in each view is considered as synthetic image corrupted by some independent noise, as shown in Figure 3(b). We use $1_{M_a}^j$ to denote the synthetic image generated by setting all elements of M_a to be 1, as shown in Figure 3(d), and call it the coverage of factor a. We use $A_{M_a}^j(M_a)$ to denote the synthetic image corresponding to a specific realization of factor a, as shown in Figure 3(e).

The potential function $\psi_a^j(M_a)$ models the similarity between the synthetic image and foreground image; the expression is as follows:

(a) Rectangles in original image (b) Detection window

Figure 2: Pedestrian detection.



(a) Original image (b) Foreground segmentation (c) Synthetic rectangle (d) The coverage of factor (e) The synthetic image of a specific realization of factor

Figure 3: Observation model of foreground detection.

$$\psi_a^j(M_a) = \begin{cases} 1 - \dfrac{\left|\left(\left(1^j_{M_a} \otimes B^j\right) \otimes \left(1^j_{M_a} - A^j_{M_a}(M_a)\right)\right) \oplus \left(\left(1^j_{M_a} - \left(1^j_{M_a} \otimes B^j\right)\right) \otimes A^j_{M_a}(M_a)\right)\right|}{\left|\left(1^j_{M_a} \otimes B^j\right) \oplus A^j_{M_a}(M_a)\right|}, & M_a \neq 0, \\ \prod_{l=1}^{L}\left(1 - \psi^j_{m^l}\left(m^l\right)\right), & M_a = 0, \end{cases} \tag{6}$$

where $\otimes$ denote pixel-wise "and" operator and $\oplus$ denote pixel-wise "or" operator, respectively. $|\cdot|$ is the sum of pixels. $M_a \neq 0$ means that the elements of factor a are not all 0, and $M_a = 0$ means that the elements of factor a are all 0. In case of $M_a = 0$, we introduce $\psi^j_{m^l}(m^l)$ to indicate the similarity between the synthetic rectangle A^j_l and the foreground observation B^j. ψ takes value between 0 and 1. If the synthetic image and the observation match perfectly, ψ takes value of 1, and vice versa.

2.3. Determination of Factor a. Factor a reflects the dependency structure of occupancy state variables due to occlusions between pedestrians. In fact, there exists very complex dependency structure among state variables. For any k and l, m^k and m^l are dependent if the intersection of their corresponding synthetic rectangles is not empty. Accordingly, we can form a factor for each location k, including k and all relevant locations. Figure 4 shows a factor under two views, consisting of k (red) and relevant locations (blue).

The complexity of the inference on the factor graph model increases exponentially with the factor size. Therefore it is intractable to use belief propagation algorithm directly for the above factor graph model as Figure 4. Based on the above consideration, we propose a novel framework, which can effectively reduce the computational complexity and ensure the accuracy of localization and tracking.

3. A Novel Framework Based on Probabilistic Inference and Data Association

As mentioned above, due to occlusion between pedestrians and overlap between camera views, the factor a may include too many variables. Thus we cannot directly do reasoning on the above factor graph. In fact, the factor graph models the occlusion between pedestrians in the ground plane where factor covers (called factor window in the following). If the size of factor window decreases, the occlusion between pedestrians in short distance is considered; if it increases, the long

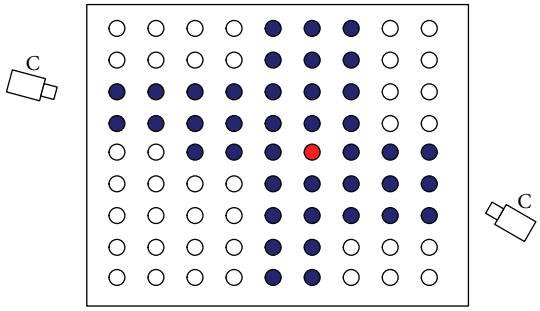


Figure 4: A typical factor-in-factor graph model.

distance is considered. Given the specific foreground image, our basic assumption is that decreasing the size of factor window makes the detection result more conservative. In other words, smaller factor windows lead to more false alarms but equal or less missing detections. Figure 5 illustrates this idea.

Figure 5 shows people detection results under different size of factor windows for single camera view. When factor window covers the whole ground, correct results are obtained by inferring based on the graph model and thresholding, as shown in Figures 5(b) and 5(c). When the size of factor window is 2×2, the ground occupancy map and the image annotation according to it are shown in Figures 5(d) and 5(e). As we can see, when the window becomes smaller, the locations on the ground occluded by pedestrians labeled with 1 and 2 are mislabeled as occupied position, leading to the occurrence of false alarms but no missed detection.

Based on the above assumption, we propose a novel framework, as shown in Figure 6. The factor window becomes larger and larger along with iterations. At each round, we first specify a fixed window size for factors and determine variables contained in each factor via sliding the factor window on the ground. Initially, we assume the size of factor window is 3×3, as $P^{(1)}$ in Figure 6.

When we calculate the probabilistic occupancy map $\mu_t^{(1)}$ of the first round by belief propagation algorithm [10] based on model $P^{(1)}$, high probability value may appear in the locations where there is nobody actually. In this case, if we threshold the probabilistic occupancy map to determine the ground occupancy map, lots of false alarms would occur. So we take the probabilistic occupancy maps $\mu_t^{(1)}$, $t = 1 : T$, as input and carry out data association by using the k-shortest paths algorithm [11] to determine object trajectory. Then we get the ground occupancy map of each time instant $\Phi_t^{(1)}$, $t = 1 : T$.

Locations marked with 1 in $\Phi_t^{(1)}$ are called suspected positions, indicating where they may be occupied by somebody. Data association takes advantage of spatiotemporal constraints of people motion and usually reduces the number of suspected positions dramatically compared with the thresholding method.

In the next round, we enlarge the size of factor window and determine variables of each factor by masking the factor window on suspected positions. Due to the expansion of the size of factor window, we can consider the dependency in longer distance, making the probabilistic occupancy map more peaky on the locations where pedestrians exist. According to $\Phi_t^{(1)}$, we determine model $P^{(2)}$ and then achieve $\mu_t^{(2)}$ by inferring on $P^{(2)}$. Taking $\mu_t^{(2)}$ as input, data association is performed to reduce the number of suspected positions, resulting in $\Phi_t^{(2)}$. The scale of the factor can be controlled to avoid combination expansion.

The iteration continues until the factor window covers the whole ground, as $P^{(3)}$, leading the peak of the probabilistic occupancy map to appear in locations where actually pedestrians exist, as $\mu_t^{(3)}$. Ground occupancy map is finally achieved through data association, as $\Phi_t^{(3)}$. Next, we will discuss how to calculate μ_t by probabilistic inference using belief propagation algorithm on factor graph model and how to calculate Φ_t by data association using k-shortest paths algorithm on cost net flow model, respectively.

3.1. Probabilistic Inference. In the sth round, we calculate the marginal probability $\mu_t^{(s)}$ of occupancy state of each suspected position according to the factor graph model $P^{(s)}$. Belief propagation algorithm aims at calculating marginal probability based on the factor graph model [10]. We first introduce messages between variable nodes and their neighboring factor nodes and vice versa. The message $m_{a\to k}(m^k)$ from the factor node a to the variable node k is a vector over the possible states of m^k. This message can be interpreted as a statement from factor node a to variable node k about the relative probabilities that k is in its different states, based on the function f_a. The message $n_{k\to a}(m^k)$ from the variable node k to the factor node a may in turn be interpreted as a statement about the relative probabilities that node k is in its different states, based on all the information that node k has except for that based on the function f_a. The messages are updated according to the following rules:

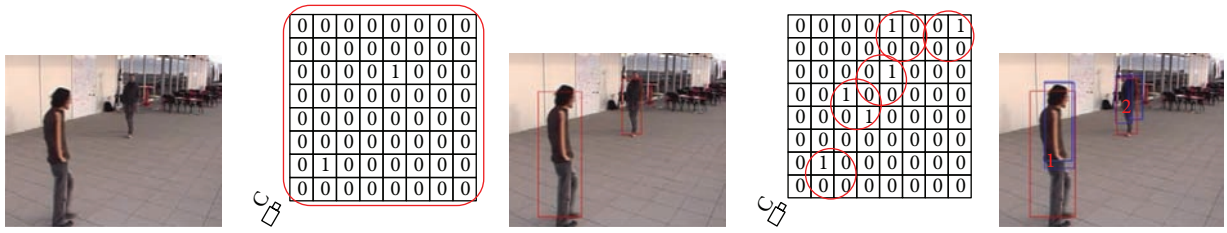
$$n_{k\to a}\left(m^k\right) := \prod_{c\in N(k)\backslash a} m_{c\to a}\left(m^k\right),$$
$$m_{a\to k}\left(m^k\right) := \sum_{M_a\backslash m^k} f_a\left(M_a\right) \prod_{l\in N(k)\backslash a} n_{l\to a}\left(m^l\right). \tag{7}$$

Here, $N(k)\backslash a$ denotes all the nodes that that are neighbors of node k except for node a, and $\sum M_a \backslash m^k$ denotes a sum over all the variables M_a that are arguments of f_a except m^k.

The messages are usually initialized to $m_{a\to k}(m^k) = 1$ and $n_{k\to a}(m^k) = 1$ for all factor nodes a, variable nodes k, and states m^k. Then we introduce the belief $b_k(m^k)$ at variable node k, which is the BP approximation to the exact marginal probability function $\mu(m^k)$. The belief $b_k(m^k)$ can be computed from

$$b_k\left(m^k\right) \propto \prod_{a\in N(k)} m_{a\to k}\left(m^k\right), \tag{8}$$

where we have used the proportionality symbol $\propto$ to indicate that one must normalize the beliefs so that they sum to one. The BP message-update equations are iterated until they (hopefully) converge; at this point the beliefs can be read off from (8).



(a) Original image (b) The truth of the ground occupancy map (c) The image annotation of the true targets (d) The ground occupancy map under 2×2 factor window (e) The image annotation according to (d)

FIGURE 5: The influence of factor window on target detection.

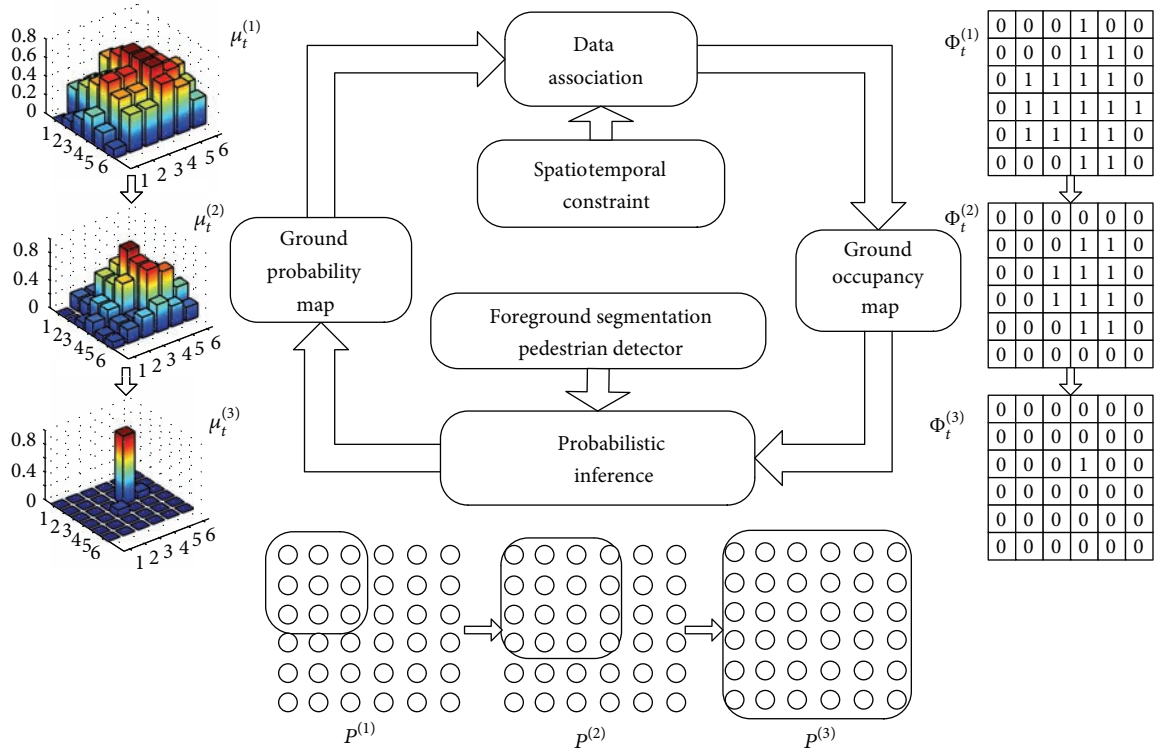


FIGURE 6: The proposed framework.

3.2. Data Association. The purpose of data association is to determine the number of pedestrians and the trajectory of each pedestrian according to probabilistic occupancy maps μ_t, $t = 1 : T$, obtained in the last iteration and the spatiotemporal constraints of pedestrian movement. We cast data association as a min-cost flow problem and follow the method in [11] to solve it. For the completeness of this paper we will sketch the basic formulation in this subsection. We refer the interested reader to [11] for details.

We take probabilistic occupancy maps μ_t, $t = 1 : T$ as input and assume people at location k at time t can only reach the neighborhood of 3×3 at time $t + 1$. Based on this constraint, we establish network flow model and transform data association to min-cost network flow problem that can be solved by k-shortest paths algorithm. The output of data association is the ground occupancy maps Φ_t, $t = 1 : T$.

To model occupancy over time, let us consider a labeled directed graph with KT vertices, which represents every location at every instant. Its edges correspond to admissible object motions, which means that there is one edge $e_t^{i,j}$ from (t, i) to $(t + 1, j)$ if and only if $j \in N(i)$; here $N(i)$ denotes the neighborhood of j, as solid lines in Figure 7.

To consider that the number of tracked objects may vary over time, we introduce two additional nodes v_{source} and $v_{\sin k}$, which are linked to all the nodes representing positions through which objects can, respectively, enter or exit the area, such as doors or borders of the camera field of view. In addition, a flow goes from v_{source} to all the nodes of the first

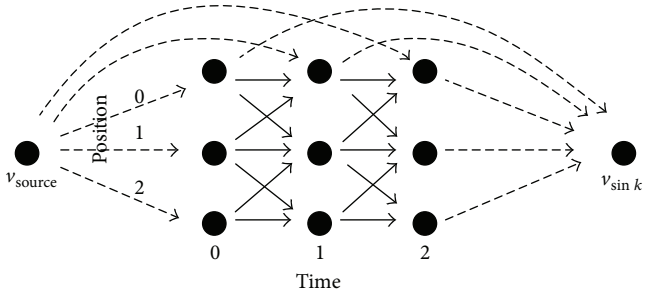


Figure 7: Network flow model.

frame, and reciprocally a flow goes from all the nodes of the last frame to $v_{\sin k}$, as dotted lines in Figure 7. The complete network flow model is shown in Figure 7.

A directed edge $e_t^{i,j}$ is assigned the cost value

$$c\left(e_t^{i,j}\right) = -\log\left(\frac{\mu_t^i}{1-\mu_t^i}\right). \tag{9}$$

The cost value of the edges emanating from the source node is set to zero to allow objects to appear at any entrance position and at any time instant at no cost. Each edge $e_t^{i,j}$ is labeled with a discrete variable $f_t^{i,j}$ standing for the number of objects. Each vertex is labeled with a discrete variable Φ_t^j (0 or 1) standing for the number of objects located at j at time t. The set of all flow variables should satisfy the following conditions (10)–(13).

For all t, the sum of flows arriving at any location j is equal to Φ_t^j, which also is the sum of outgoing flows from location j at time t. We must therefore have

$$\forall t, j, \quad \underbrace{\sum_{i,j\in N(k)} f_{t-1}^{i,j}}_{\text{Arriving at } j \text{ at } t} = \Phi_t^j = \underbrace{\sum_{k\in N(j)} f_t^{j,k}}_{\text{Leaving from } j \text{ at } t}. \tag{10}$$

Furthermore, since a location cannot be occupied by more than one object at a time, we can set an upper bound of 1 to the sum of all outgoing flows from a given location and impose

$$\forall t, j, \quad \sum_{k\in N(j)} f_t^{j,k} \le 1. \tag{11}$$

A similar constraint applies to the incoming flows, but we do not need to explicitly state it, since it is implicitly enforced by (9). Finally, the flows have to be nonnegative and we have

$$\forall k, t, j, \quad \sum f_t^{k,j} \ge 0. \tag{12}$$

Finally, we introduce an additional constraint that ensures that all flows departing from v_{source} eventually end up in $v_{\sin k}$:

$$\underbrace{\sum_{j\in N(v_{\text{source}})} f_t^{v_{\text{source}},j}}_{\text{Leaving } v_{\text{source}}} = \underbrace{\sum_{k:v_{\sin k}\in N(k)} f_t^{k,v_{\sin k}}}_{\text{Arriving } v_{\sin k}}. \tag{13}$$

Under the conditions of (10)–(13), data association problem is converted into minimum cost flow problem and can be written as

$$f^* = \arg\min_{f\in\mathfrak{F}} \sum_{t,i} c\left(e_t^{i,j}\right) \sum_{j\in N(i)} f_t^{i,j}. \tag{14}$$

Here, $\mathfrak{F}$ denotes the set of feasible solutions satisfying conditions (10)–(13). And because the occupancy variable Φ_t^i can be expressed as

$$\forall t, i, \quad M_t^i = \sum_{j\in N(i)} f_t^{i,j}. \tag{15}$$

As a result, flow variable $f_t^{i,j}$ indirectly provides occupancy variable Φ_t^i. The min-cost flow problem has very efficient off-the-self algorithm, and we adopt the k-shortest paths algorithm—given a pair of nodes, namely, v_{source} and $v_{\sin k}$, in a graph G, find the k paths $\{p_1, \ldots, p_k\}$ between these nodes such that the total cost of the paths is minimum.

4. Results

4.1. Test Data. Our data set consists of Terrace sequence, Basketball sequence, Passageway sequence [14] and PETS 2009 S2/L1 sequence [15]. Terrace sequence is 3 1/2-minute outdoor sequence consisting of up to 9 people appearing one after the other and walking in front of the cameras. It tests the ability of our algorithm to cope with a moderately crowded environment. Basketball sequence involves 10 players and 1 referee in a game on half a Basketball court. It is a difficult sequence with fast moving people and many occlusions. Passageway sequence involves 7 people passing through a public underground Passageway. This is very challenging for several reasons. First, lighting conditions are very poor, which is typical in real-world surveillance situations. A large portion of the images is either underexposed or saturated. Second, the area covered by the system is large, and people get very small on the far end of the scene, making their precise localization challenging. Finally, large parts of the scene are filmed by only two cameras or even a single camera. PETS 2009 sequence filmed at a road corner of a university campus involves about 10 people, due to the large monitored area, and part of the scene is only filmed by one camera, making it a certain difficulty. In the first three pedestrian environments, the scene was filmed by 4 DV cameras with overlapping fields of view, each of which is placed in a corner of the monitored area. The video format is DV PAL, downsampled to 360 $*$ 288 pixels at 25 fps, and the 4 video streams were synchronized manually after data acquisition. The PETS 2009 sequence was filmed by seven cameras: three dedicated video surveillance cameras and four DV cameras. The DV cameras were placed at about 2 meters above the ground, whereas the video surveillance cameras were located between 3 and 5 meters above it and significantly farther from the scene. The frame rate for all cameras was set to 7 fps and downsampled to 720 $*$ 576 pixels. Due to calibration imprecision, only five out of the seven available camera views were used for people detection. We select four DV cameras (Table 1).

Table 1: Datasets used for test.

Sequence	Frame	Dimensions (m)	Grid size (cm)	Locations
Terrace	5250	9 ∗ 6	30 ∗ 30	714
Basketball	9350	15 ∗ 14	30 ∗ 30	2350
Passageway	2500	12 ∗ 30	40 ∗ 40	4000
PETS 2009	795	18.5 ∗ 20	35 ∗ 35	3000

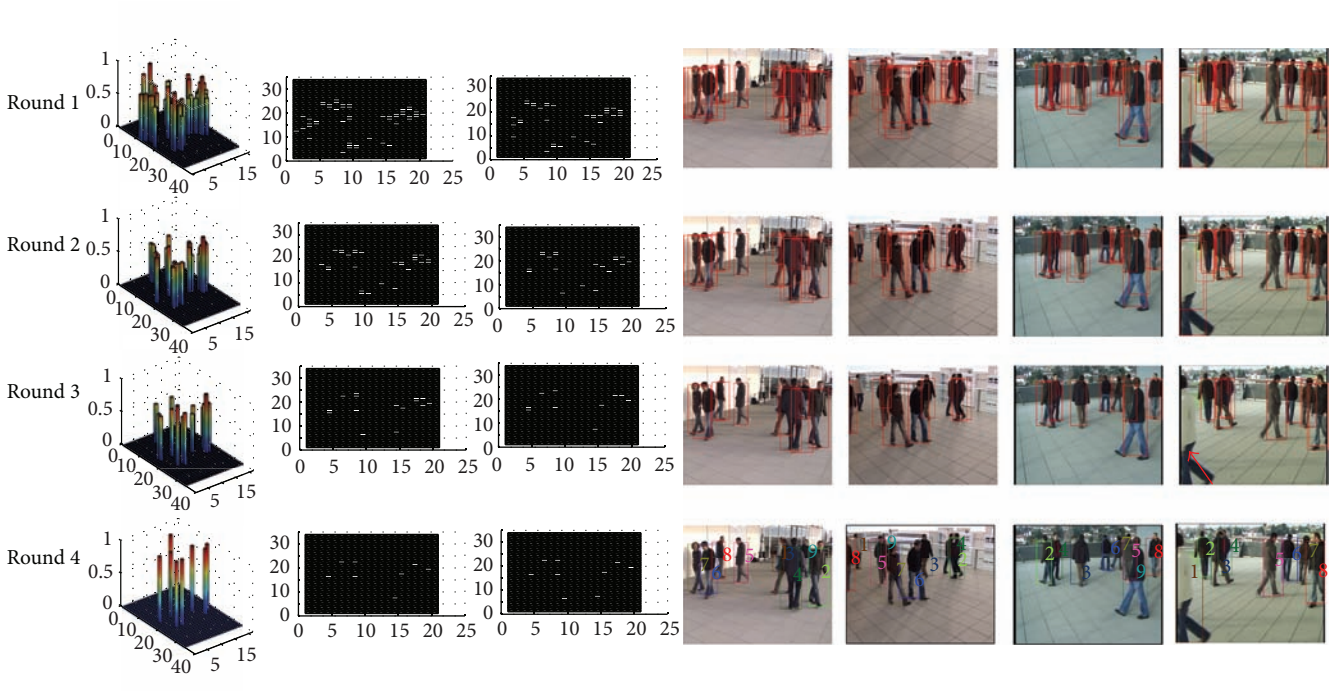


Figure 8: A typical case illustrating our proposed framework.

4.2. Implementation Details. In foreground segmentation, we adopt the method proposed in [16] to obtain binary image; in pedestrian detection, HOG feature is calculated by method proposed in [17] and the kernel function of support vector machine is linear, realized by LIBSVM tools [18]. For Basketball datasets, the trained model of INRIA datasets [19] cannot achieve good performance due to the rapid motion of players, so we add extra data depicted in [20] to retrain more robust model. In probabilistic inference, the manner of factor window enlargement is very flexible. In order to speed up the algorithm, the factor window is enlarged in an adaptive manner. In each round, we choose the maximum factor window size ensuring that the number of unresolved states in each individual factor does not exceed 8.

Our original formulation is not applicable to cases where the numbers of locations in the ground and frames in the video are very large. In Basketball sequence, the resulting cost-flow network consists of about 21 million nodes and 197 million edges. This is beyond the memory capability of common PC. So in our experiments, following [6], we process the video sequence by batches of 200 frames instead of the whole sequence and keep the results of the first 20 frames and slide our temporal window to achieve consistency over successive batches.

4.3. Case Study. The frame 3250 of Terrace sequence is chosen as a typical case to illustrate the process of the framework proposed in this paper.

Each row in Figure 8 shows the result of a single round. The first column represents the probabilistic occupancy map given by belief propagation algorithm; the second column represents the ground occupancy map obtained by thresholding the probabilistic occupancy map; the third column represents the ground occupancy map achieved by k-shortest paths algorithm; the last four columns represent annotations in different views according to the output of k-shortest paths algorithm. Because the outputs of k-shortest paths algorithm are tracklets until the last round, annotations in the first three rows are not specified to targets, highlighted with red. However, annotations in last row show the final result of tracking, so we highlight different targets with different colors.

We can see from the second and third columns in Figure 8 that the true occupied positions can be detected by both thresholding and data association, but the number of suspected positions in the latter is less than the former. This proves that data association using spatiotemporal constraints can reduce false alarms. Consequently, taking such ground occupancy map as input for the next round can greatly improve the speed of inference.

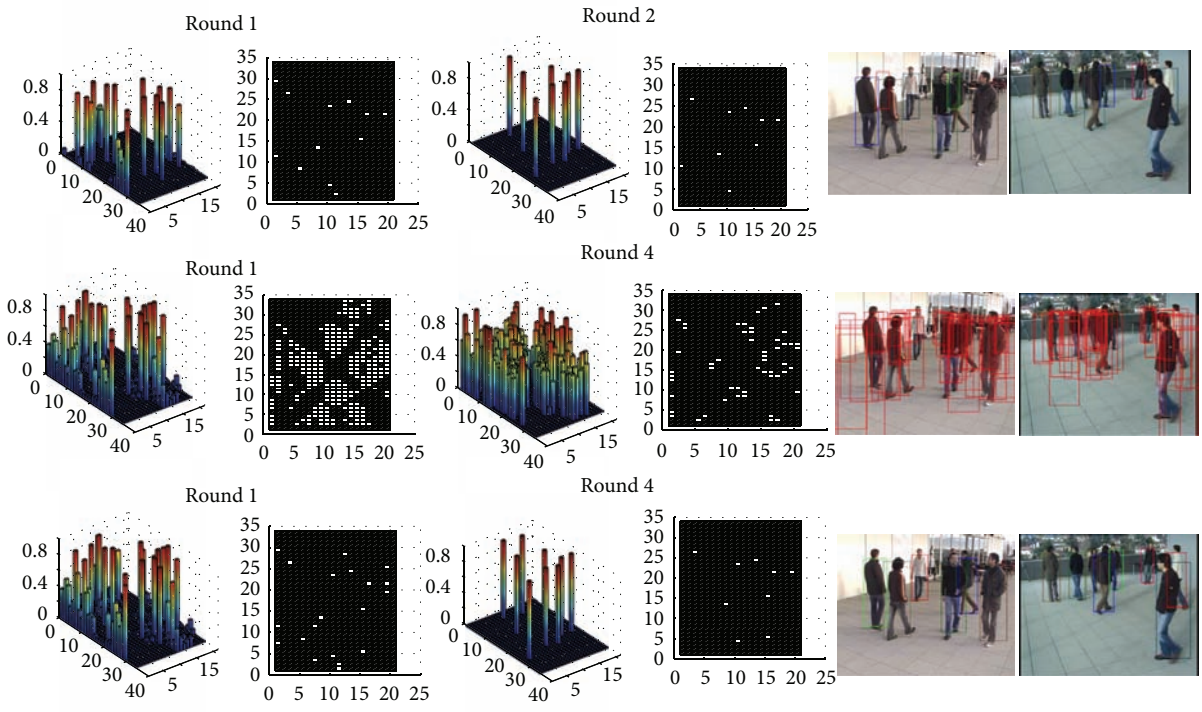


Figure 9: Illustration of the effect of pedestrian detection.

In the first round, we set the size of factor window to 3 × 3. Thus we cannot handle occlusions in long distance. As a result, the probabilistic occupancy map is relatively flat and the number of suspected positions is large. In the following round, the factor window is enlarged in an adaptive manner. Thus the occlusions in longer distance are considered and the resulting probabilistic occupancy map is more peaked. Moreover, the number of trajectories given by k-shortest paths algorithm is reduced and the length of each trajectory is extended. In our framework, localization and data association are performed alternatively until factor window covers the whole ground. In the final probabilistic occupancy maps, the peaks present in few locations which are occupied by people actually.

Now we focus on the red arrow pointing to the people labeled with 1. In the third iteration, this target is lost due to the large number of false alarms around it in the preceding and following frames. However, after the enlargement of factor window in the fourth iteration, these false alarms are deleted because of the consideration of occlusions in larger area. Consequently, target 1 is recovered by data association using spatiotemporal constraints.

4.4. Effect of Pedestrian Detection. It is interesting to ask to what extent does our method rely on the suspected position removing effect of data association. In our experiments, we find that the behavior of data association relies heavily on the flatness of probabilistic occupancy map. If the probabilistic occupancy map is flat, k-shortest paths algorithm usually produces a large number of short tracks, and the number of suspected positions decreases very slowly. If the probabilistic occupancy map is peaky, k-shortest paths algorithm usually produces a few long tracks, reducing the number of suspected positions rapidly.

We find that the incorporation of the pedestrian detector into the potential function of factor graph is critical to the performance of our method. Figure 9 illustrates the effect of pedestrian detection. The first row shows the resulting probabilistic occupancy map and ground occupancy map at each round when ω_1 and ω_2 in (2) are set to be 1 and 0, respectively; that is, only pedestrian detection information is used. We can see from Figure 9 that, under this setting, the final solution can be reached rapidly using our method. However, false alarms and missing detection occur because the factor graph model under this setting cannot handle the occlusion between persons. The second row in Figure 9 shows the results when ω_1 and ω_2 are set to be 0 and 1, respectively; that is, only foreground detection information is used. Under this setting, the number of suspected positions decreases very slowly because the probabilistic occupancy maps obtained at earlier iterations are rather flat. The third row in Figure 9 shows the results when both ω_1 and ω_2 are set to be 0.5; that is, we use the combination of pedestrian detection and foreground detection. As we can see, our method can find the correct solution in this case. Under this setting, our method combines the discriminative power of pedestrian detector and the ability to handle occlusions of foreground mask based inference.

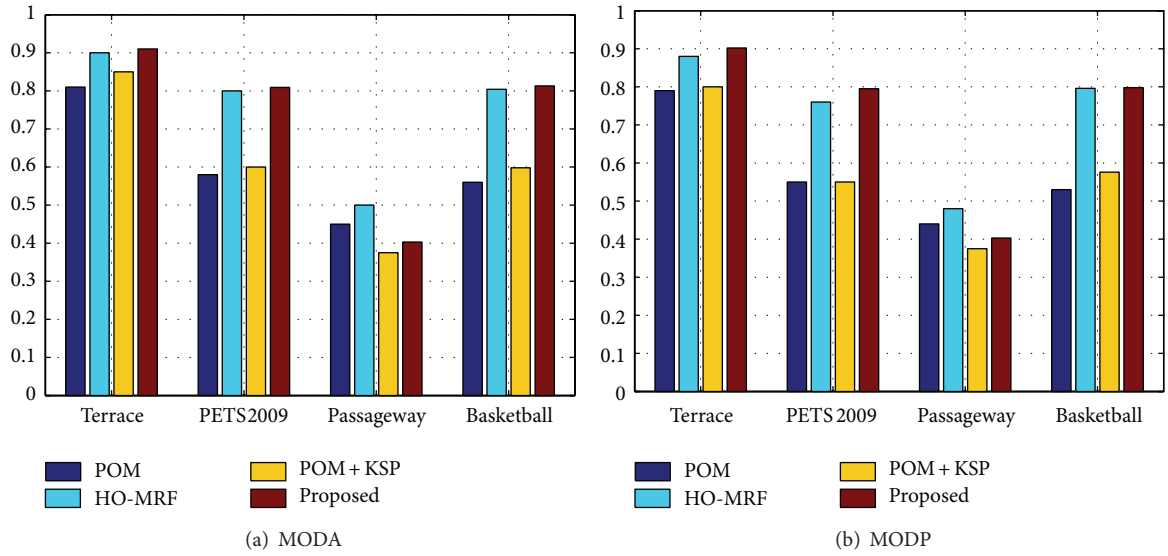


Figure 10: Detection metrics.

In some extremely difficult cases, for example, very crowded group of people moving in a small region, the output probabilistic occupancy map may be rather flat and the data association cannot be able to remove the suspected position effectively. In this case, our framework reduces to the method in [7]. However, in our experiments we find our method can always remove the suspected position effectively by proper combination of information from pedestrian detector and foreground masks.

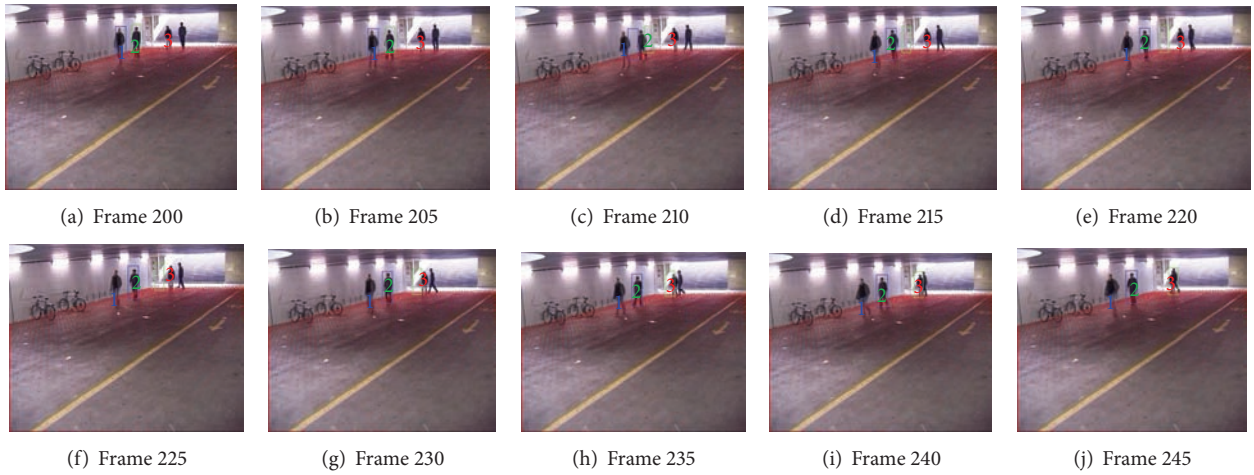
4.5. Baseline and Evaluation Metrics. In order to demonstrate the effectiveness of our algorithm, we compare the following four algorithms: probabilistic occupancy map (POM) proposed in [6]; cascaded optimization on higher-order MRFs (HO-MRF) proposed in [7]; k-shortest paths (KSP) proposed in [11]; and the proposed framework in this paper.

We adopt the commonly used detection metrics, multiple object detection accuracy (MODA), multiple object detection precision (MODP) and tracking metrics multiple object tracking accuracy (MOTA), and multiple object tacking precision (MOTP) [21]. MODA takes into account false alarm, missed detections. MODP measures the spatial overlap information between the ground truth and the detections. MOTA takes into account false alarm, missed detections, and identity switches. MOTP measures the average distance between ground truth trajectories and the system-generated trajectories. The higher the above metrics are, the better the performance of the algorithm is.

4.6. Comparison Results. Figure 10 shows detection metrics of the above algorithms. On all sequences, HO-MRF and our proposed method are better than POM and POM + KSP in terms of MODA and MODP because of the modeling power of factor graph model. POM approximates the joint posterior distribution of occupancy states by a product law minimizing the KL divergence; therefore the performance of detection deteriorates in case of dense crowd or serious occlusions; POM + KSP is slightly better than POM, because POM + KSP takes the output of POM as input and takes advantage of spatiotemporal constraints, leading to reduction of false alarms and missed detections. By using factor graph (or MRF) model with varying-sized factors (or cliques), our method and HO-MRF can deal with occlusions in large area, leading to better detection results. Ideally, if none of the true occupied positions were misdeleted during removing of suspected positions, upon convergence HO-MRF and our method could reach the equivalent result to that given by optimizing directly on the joint distribution of occupancy states, which is usually computationally intractable. Actually, we find in our experiments that true occupied positions are seldom misdeleted in HO-MRF and our method during removing of suspected positions. This can be verified in Figure 13 that the FN metrics of HO-MRF and our method are always equal or less than that of POM and POM + KSP. In contrast to HO-MRF which makes hard decision through optimization on MRF model at each iteration, our method takes the soft occupancy probabilistic map as input and exploits spatiotemporal information to find the ground occupancy map and thus can reduce the suspected positions more efficiently and effectively. Consequently, the number of suspected positions decreases much faster in our method than in HO-MRF, leading to significant improvement in speed.

It is noticeable that for Passageway sequence, POM + KSP and our method, which exploit the spatiotemporal constraints, perform worse than POM and HO-MRF. As mentioned above, the observing condition in Passageway sequence is rather poor, resulting in large number of false alarms and missing detections, which usually occurs continuously. Utilization of spatiotemporal constraints in this case, however, may further deteriorate the performance of detection. Figure 11 shows a typical case.

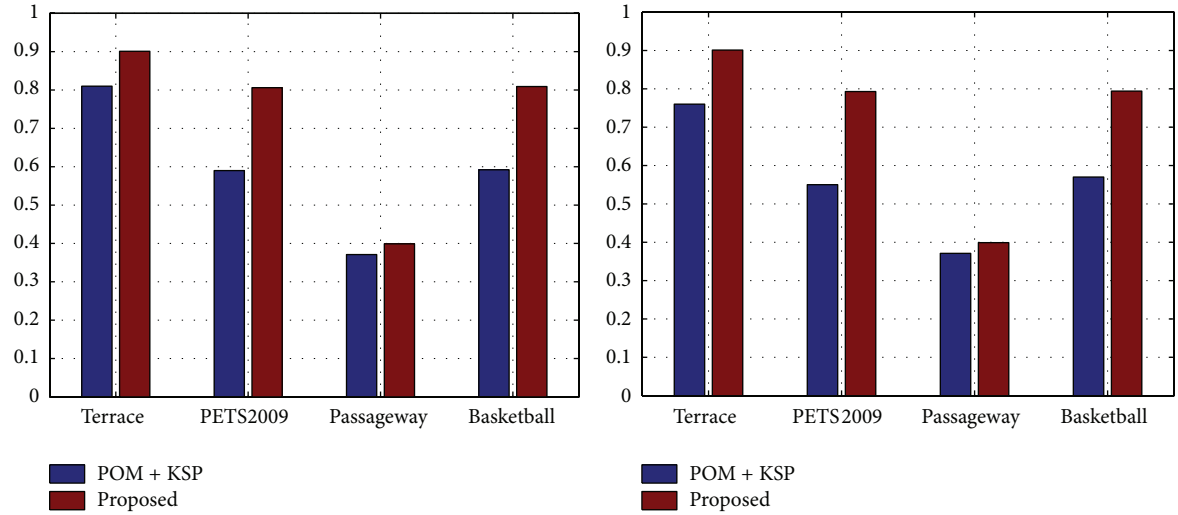
As we can see from Figure 11, at the beginning, target 1 and target 2 are detected in frame 200 and assigned with blue box



(a) Frame 200 (b) Frame 205 (c) Frame 210 (d) Frame 215 (e) Frame 220

(f) Frame 225 (g) Frame 230 (h) Frame 235 (i) Frame 240 (j) Frame 245

Figure 11: Identity switches of targets.



(a) MOTA (b) MOTP

Figure 12: Tracking metrics.

and green box, respectively. During the period starting from frame 205 to frame 225, the occupancy probabilities of corresponding target 1 are small due to the poor lighting conditions. This leads to the association of target 1's track with false alarms in the neighborhood of target 1, as shown by the blue box in frames 205~210. At frame 215, the blue box is assigned to target 2. Because of the hard constraint in KSP that a single position can be associated with only one track, the green box originally assigned to target 2 is moving to target 3, as shown in frames 210~240. After frame 225, the targets 1 and 2 are close to the camera and the calculated occupancy probability maps are much reliable. However, target 1 cannot be assigned to any track until another target moves into its neighborhood. This is mainly because the track corresponding to target 1, that is, the blue box, is now associated with target 2. And at the same time, in our network model the virtual nodes v_{source} and $v_{\text{sin}\,k}$ are connected only with positions in the entrance region of the Passageway. In other words, trajectory can only start or finish near the entrance of Passageway. Therefore, KSP cannot generate a new track for target 1 because target 1 is too far from the entrance, resulting in a lot of missing detections.

Figure 12 shows tracking metrics of POM + KSP and our method on different sequences. We can see that our method is superior to POM + KSP on all sequences in terms of MOTA and MOTP. There are two main reasons. (1) POM + KSP follows the paradigm of detection-tracking, so detection results directly determine the performance of tracking. In order to avoid the computational intractability in the joint state space, POM approximates the probability of occupancy on each individual location by the marginal of a product law which minimizes the KL divergence from the posterior joint distribution. In contrast, by alternating between detection and tracking, our method can deal with the whole state space by enlarging the factor size sequentially and in principle

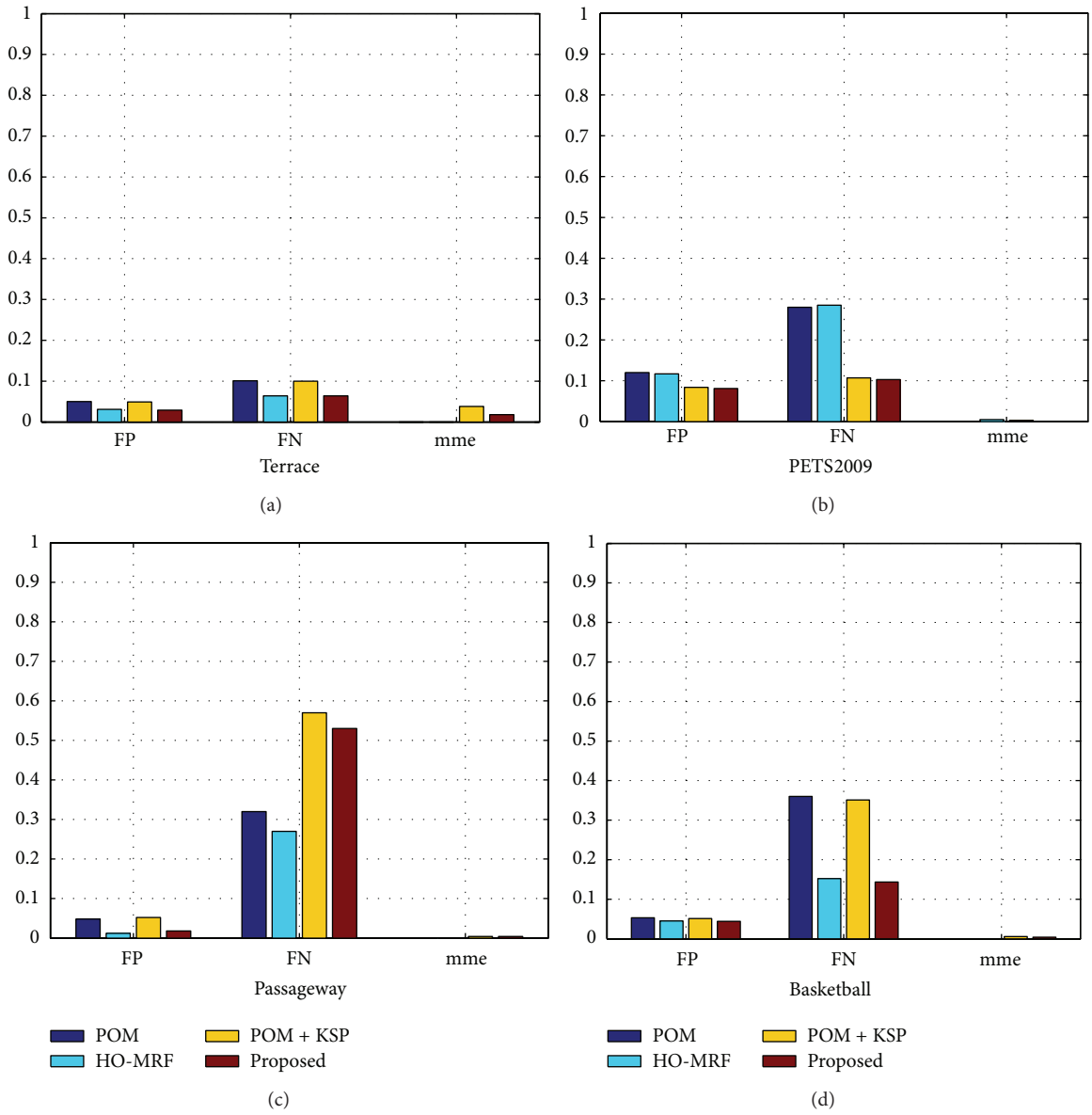


Figure 13: Components of tracking metrics.

is able to find the global optimal solution with reasonable computation provided that no misdeletion of true occupied positions occurs. (2) POM utilizes foreground information only, while our method combines the information from foreground detection and pedestrian detection in a principle way by defining a novel potential function, thanks to the modeling power of factor graph model.

Figure 14 shows some examples of pedestrian detection and tracking results in our experiments. Rows 1 and 2 come from the Terrace sequence; rows 3 and 4 come from the PETS 2009 sequence; rows 5 and 6 come from the Passageway sequence; rows 7 and 8 come from the Basketball sequence. Here, the odd rows are the results of POM + KSP, and even rows are the results of ours.

In Terrace sequence, we can see persons labeled 6 and 7 are occluded seriously in views 1, 2, and 4. POM + KSP fails to detect them and produces a false alarm near person 6. In contrast, our method locates all the 7 persons successfully without producing any false alarm. In PETS 2009 sequence, there is a period during which two pedestrians labeled 5 and 6 walk in parallel and can only be seen in one view. This is a very difficult scenario for detection and tracking, and POM + KSP loses the tracks of these two persons completely during this period. But our method can still track them. In Passageway sequence, the foreground segmentation is unreliable due to the underexposed or saturated image. As a result, in the frame shown in Figure 12, POM + KSP takes a false alarm as detection of person 3 and misses the true person. In contrast, our method can locate person 3 correctly. In Basketball sequence, due to the complex occlusion and severe illumination, POM + KSP produces a lot of false alarms and fails to detect persons labeled 6, 7, 8, and 9. Our method takes advantage of the factor

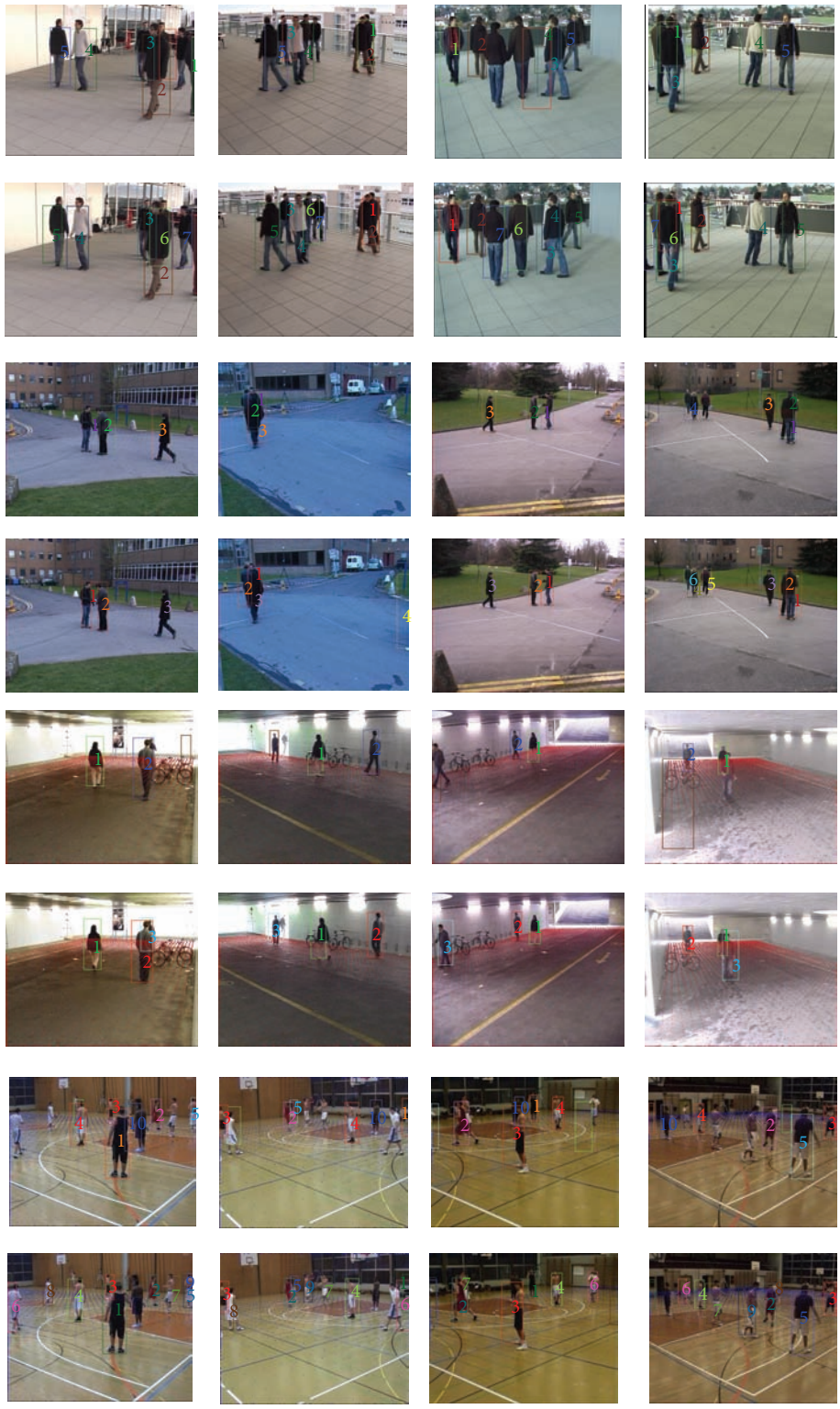


Figure 14: Tracking examples.

graph model and pedestrian detector, improving the accuracy of detection and tracking significantly.

4.7. Run Time. We implement HO-MRF and the proposed algorithm with Matlab code in common PC and use the code provided by the authors of [6, 11] for implementing POM and POM + KSP. The runtime is varying with the length of sequence and the number of locations. We report the average time for processing a single frame of each sequence, as shown in Table 2.

Table 2: Run time (s/frame).

Sequence	POM	HO-MRF	POM + KSP	Proposed
Terrace	0.5	300	1.5	30
PETS 2009	1	420	2	90
Passageway	1.5	480	2.5	120
Basketball	1	360	2	60

The probabilistic inference of HO-MRF and our method take long time due to complicated dependency structure between locations. However, by alternating detection and data association, our method reduces the number of suspect positions significantly and speeds up the detection. Thus our method achieves good compromise between accuracy and speed. We note here that POM and POM + KSP are implemented with highly optimized C code, which is very efficient in computing. The speed of our algorithm can be improved by optimizing the coding.

4.8. Limitations. In detection phase, we choose HOG detector, which represents the contour information of the whole detection window. The performance of HOG detector is not ideal in the case of crowd or serious occlusions. However, the detection result is still reliable because of the fusion of multiple visual observations. Nevertheless, if pedestrian detector based on deformable part model is adopted, occlusion can be dealt with more effectively, and detection accuracy can be further improved.

In tracking phase, we care about not only the location of targets, but also the identity of different targets. In this paper, we associate with the whole sequence to achieve the global optimal solution for multiple target trajectories. But identity switches still exist due to undistinguished observation between pedestrians in case of dense crowd or frequent occlusions. In order to solve this problem, establishing distinguished appearance model may be helpful.

5. Conclusions

This paper focuses on people tracking using camera networks with overlapping views and proposes a novel framework alternating localization and data association. The localization is realized by probabilistic inference on factor graph model through belief propagation algorithm. It makes modeling the likelihood of the entire image possible and thus avoids nonmaximum suppression. Data association is converted to min-cost flow problem solved by k-shortest paths algorithm. It enables us to search global optimal solution for multiple target trajectories. Due to enlargement of factor window round by round, the algorithm fully considers dependency between occupancy states in the whole ground while avoids computational complexity. The experiments on different datasets show that alternating localization and data association can significantly improve detection and tracking performance compared to the results of track by detection approaches. How to realize accurate detection for localization and how to exploit higher-order motion constraints and appearance model for data association will be investigated in our future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by the National Natural Science Foundation of China, under Grant no. 61174020.

References

[1] K. Kim and L. S. Davis, "Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering," in *Computer Vision—ECCV 2006: Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006, Part III*, vol. 3953 of *Lecture Notes in Computer Science*, pp. 98–109, Springer, Berlin, Germany, 2006.

[2] W. Du and J. Piater, "Multi-camera people tracking by collaborative particle filters and principal axis-based integration," in *Computer Vision—ACCV 2007: 8th Asian Conference on Computer Vision, Tokyo, Japan, November 18–22, 2007, Proceedings, Part I*, vol. 4843 of *Lecture Notes in Computer Science*, pp. 365–374, Springer, Berlin, Germany, 2007.

[3] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 505–519, 2009.

[4] 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.

[5] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1198–1211, 2008.

[6] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 267–282, 2008.

[7] W. Jiuqing and Z. Fan, "Multi-camera people localization via cascaded optimization on higher-order MRFs," in *Proceedings of the 6th International Conference on Distributed Smart Cameras (ICDSC '12)*, pp. 1–7, IEEE, 2012.

[8] B. Leibe, K. Schindler, N. Cornelis, and L. van Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1683–1698, 2008.

[9] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling detection and data association for multiple object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1948–1955, IEEE, June 2012.

[10] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[11] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, IEEE, June 2005.

[13] H.-T. Lin, C.-J. Lin, and R. C. Weng, "A note on Platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, no. 3, pp. 267–276, 2007.

[14] Multi-camera pedestrians video, "EPFL" data set: multi-camera pedestrian videos, http://cvlab.epfl.ch/data/pom.

[15] Proceedings of the 11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS '09), 2009, http://cs.binghamton.edu/~mrldata/pets2009.

[16] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, no. 3, pp. 187–193, 1995.

[17] M. J. Jones and P. Viola, "Robust real-time object detection," in *Proceedings of the Workshop on Statistical and Computational Theories of Vision*, p. 266, British Columbia, Canada, July 2001.

[18] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[19] "INRIA Person Dataset," http://pascal.inrialpes.fr/data/human/.

[20] H. Ben Shitrit, M. Raca, F. Fleuret et al., *Tracking Multiple Players Using a Single Camera*, Springer, 2013.

[21] R. Kasturi, D. Goldgof, P. Soundararajan et al., "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319–336, 2009.