Multicamera People Tracking with a Probabilistic Occupancy Map

François Fleuret, Jérôme Berclaz, Richard Lengagne, and Pascal Fua, *Senior Member*, *IEEE*

Abstract—Given two to four synchronized video streams taken at eye level and from different angles, we show that we can effectively combine a generative model with dynamic programming to accurately follow up to six individuals across thousands of frames in spite of significant occlusions and lighting changes. In addition, we also derive metrically accurate trajectories for each of them. Our contribution is twofold. First, we demonstrate that our generative model can effectively handle occlusions in each time frame independently, even when the only data available comes from the output of a simple background subtraction algorithm and when the number of individuals is unknown a priori. Second, we show that multiperson tracking can be reliably achieved by processing individual trajectories separately over long sequences, provided that a reasonable heuristic is used to rank these individuals and that we avoid confusing them with one another.

Index Terms—Multipeople tracking, multicamera, visual surveillance, probabilistic occupancy map, dynamic programming, Hidden Markov Model.

1 Introduction

In this paper, we address the problem of keeping track of people who occlude each other using a small number of synchronized videos such as those depicted in Fig. 1, which were taken at head level and from very different angles. This is important because this kind of setup is very common for applications such as video surveillance in public places.

To this end, we have developed a mathematical framework that allows us to combine a robust approach to estimating the probabilities of occupancy of the ground plane at individual time steps with dynamic programming to track people over time. This results in a fully automated system that can track up to six people in a room for several minutes by using only four cameras, without producing any false positives or false negatives in spite of severe occlusions and lighting variations. As shown in Fig. 2, our system also provides location estimates that are accurate to within a few tens of centimeters, and there is no measurable performance decrease if as many as 20 percent of the images are lost and only a small one if 30 percent are. This involves two algorithmic steps:

1. We estimate the probabilities of occupancy of the ground plane, given the binary images obtained from the input images via background subtraction [7]. At this stage, the algorithm only takes into account images acquired at the *same time*. Its basic ingredient is a generative model that represents humans as simple rectangles that it uses to create synthetic ideal images that we would observe if people were at given locations. Under this model of the images, given the true occupancy, we approximate the probabilities of occupancy at every location as the marginals of a product law minimizing the Kullback-Leibler divergence from the "true" conditional posterior distribution. This allows us to evaluate the probabilities of occupancy at every location as the fixed point of a large system of equations.
2. We then combine these probabilities with a color and a motion model and use the Viterbi algorithm to accurately follow individuals across thousands of frames [3]. To avoid the combinatorial explosion that would result from explicitly dealing with the joint posterior distribution of the locations of individuals in each frame over a fine discretization, we use a greedy approach: we process trajectories individually over sequences that are long enough so that using a reasonable heuristic to choose the order in which they are processed is sufficient to avoid confusing people with each other.

In contrast to most state-of-the-art algorithms that recursively update estimates from frame to frame and may therefore fail catastrophically if difficult conditions persist over several consecutive frames, our algorithm can handle such situations since it computes the global optima of scores summed over many frames. This is what gives it the robustness that Fig. 2 demonstrates.

In short, we combine a mathematically well-founded generative model that works in each frame individually with a simple approach to global optimization. This yields excellent performance by using basic color and motion models that could be further improved. Our contribution is therefore twofold. First, we demonstrate that a generative model can effectively handle occlusions at each time frame independently, even when the input data is of very poor quality, and is therefore easy to obtain. Second, we show that multiperson tracking can be reliably achieved by processing individual trajectories separately over long sequences.

- *F. Fleuret, J. Berclaz, and P. Fua are with the École Polytechnique Fédérale de Lausanne, Station 14, CH-1015 Lausanne, Switzerland. E-mail: {francois.fleuret, jerome.berclaz, pascal.fua}@epfl.ch.*
- *R. Lengagne is with GE Security-VisioWave, Route de la Pierre 22, 1024 Ecublens, Switzerland. E-mail: richard.lengagne@ge.com.*

Manuscript received 14 July 2006; revised 19 Jan. 2007; accepted 28 Mar. 2007; published online 15 May 2007.
Recommended for acceptance by S. Sclaroff.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0521-0706.
Digital Object Identifier no. 10.1109/TPAMI.2007.1174.

Fig. 1. Images from two indoor and two outdoor multicamera video sequences that we use for our experiments. At each time step, we draw a box around people that we detect and assign to them an ID number that follows them throughout the sequence.

In the remainder of the paper, we first briefly review related works. We then formulate our problem as estimating the most probable state of a hidden Markov process and propose a model of the visible signal based on an estimate of an occupancy map in every time frame. Finally, we present our results on several long sequences.

2 Related Work

State-of-the-art methods can be divided into monocular and multiview approaches that we briefly review in this section.

2.1 Monocular Approaches

Monocular approaches rely on the input of a single camera to perform tracking. These methods provide a simple and easy-to-deploy setup but must compensate for the lack of 3D information in a single camera view.

2.1.1 Blob-Based Methods

Many algorithms rely on binary blobs extracted from single video [10], [5], [11]. They combine shape analysis and tracking to locate people and maintain appearance models in order to track them, even in the presence of occlusions. The *Bayesian Multiple-BLob tracker (BraMBLe)* system [12], for example, is a multiblob tracker that generates a blob-likelihood based on a known background model and appearance models of the tracked people. It then uses a particle filter to implement the tracking for an unknown number of people.

Approaches that track in a single view prior to computing correspondences across views extend this approach to multi camera setups. However, we view them as falling into the same category because they do not simultaneously exploit the information from multiple views. In [15], the limits of the field of view of each camera are computed in every other camera from motion information. When a person becomes visible in one camera, the system automatically searches for him in other views where he should be visible. In [4], a background/foreground segmentation is performed on calibrated images, followed by human shape extraction from foreground objects and feature point selection extraction. Feature points are tracked in a single view, and the system switches to another view when the current camera no longer has a good view of the person.

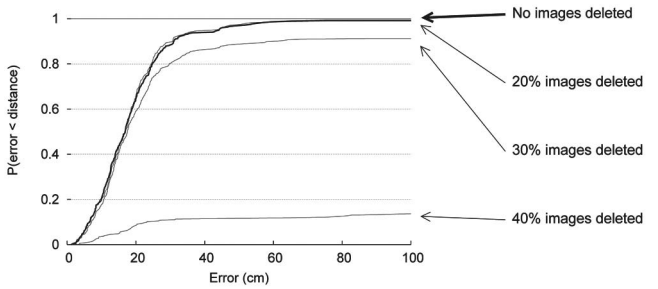

Fig. 2. Cumulative distributions of the position estimate error on a 3,800-frame sequence (see Section 6.4.1 for details).

2.1.2 Color-Based Methods

Tracking performance can be significantly increased by taking color into account.

As shown in [6], the mean-shift pursuit technique based on a dissimilarity measure of color distributions can accurately track deformable objects in real time and in a monocular context. In [16], the images are segmented pixelwise into different classes, thus modeling people by continuously updated Gaussian mixtures. A standard tracking process is then performed using a Bayesian framework, which helps keep track of people, even when there are occlusions. In such a case, models of persons in front keep being updated, whereas the system stops updating occluded ones, which may cause trouble if their appearances have changed noticeably when they re-emerge.

More recently, multiple humans have been simultaneously detected and tracked in crowded scenes [20] by using Monte-Carlo-based methods to estimate their number and positions. In [23], multiple people are also detected and tracked in front of complex backgrounds by using mixture particle filters guided by people models learned by boosting. In [9], multicue 3D object tracking is addressed by combining particle-filter-based Bayesian tracking and detection using learned spatiotemporal shapes. This approach leads to impressive results but requires shape, texture, and image depth information as input. Finally, Smith et al. [25] propose a particle-filtering scheme that relies on Markov chain Monte Carlo (MCMC) optimization to handle entrances and departures. It also introduces a finer modeling of interactions between individuals as a product of pairwise potentials.

2.2 Multiview Approaches

Despite the effectiveness of such methods, the use of multiple cameras soon becomes necessary when one wishes to accurately detect and track multiple people and compute their precise 3D locations in a complex environment. Occlusion handling is facilitated by using two sets of stereo color cameras [14]. However, in most approaches that only take a set of 2D views as input, occlusion is mainly handled by imposing temporal consistency in terms of a motion model, be it Kalman filtering or more general Markov models. As a result, these approaches may not always be able to recover if the process starts diverging.

2.2.1 Blob-Based Methods

In [19], Kalman filtering is applied on 3D points obtained by fusing in a least squares sense the image-to-world projections of points belonging to binary blobs. Similarly, in [1], a Kalman filter is used to simultaneously track in 2D and 3D, and object

locations are estimated through trajectory prediction during occlusion.

In [8], a best hypothesis and a multiple-hypotheses approaches are compared to find people tracks from 3D locations obtained from foreground binary blobs extracted from multiple calibrated views.

In [21], a recursive Bayesian estimation approach is used to deal with occlusions while tracking multiple people in multiview. The algorithm tracks objects located in the intersections of 2D visual angles, which are extracted from silhouettes obtained from different fixed views. When occlusion ambiguities occur, multiple occlusion hypotheses are generated, given predicted object states and previous hypotheses, and tested using a branch-and-merge strategy. The proposed framework is implemented using a customized particle filter to represent the distribution of object states.

Recently, Morariu and Camps [17] proposed a method based on dimensionality reduction to learn a correspondence between the appearance of pedestrians across several views. This approach is able to cope with the severe occlusion in one view by exploiting the appearance of the same pedestrian on another view and the consistence across views.

2.2.2 Color-Based Methods

Mittal and Davis [18] propose a system that segments, detects, and tracks multiple people in a scene by using a wide-baseline setup of up to 16 synchronized cameras. Intensity information is directly used to perform single-view pixel classification and match similarly labeled regions across views to derive 3D people locations. Occlusion analysis is performed in two ways: First, during pixel classification, the computation of prior probabilities takes occlusion into account. Second, evidence is gathered across cameras to compute a presence likelihood map on the ground plane that accounts for the visibility of each ground plane point in each view. Ground plane locations are then tracked over time by using a Kalman filter.

In [13], individuals are tracked both in image planes and top view. The 2D and 3D positions of each individual are computed so as to maximize a joint probability defined as the product of a color-based appearance model and 2D and 3D motion models derived from a Kalman filter.

2.2.3 Occupancy Map Methods

Recent techniques explicitly use a discretized occupancy map into which the objects detected in the camera images are back-projected. In [2], the authors rely on a standard detection of stereo disparities, which increase counters associated to square areas on the ground. A mixture of Gaussians is fitted to the resulting score map to estimate the likely location of individuals. This estimate is combined with a Kallman filter to model the motion.

In [26], the occupancy map is computed with a standard visual hull procedure. One originality of the approach is to keep for each resulting connex component an upper and lower bound on the number of objects that it can contain. Based on motion consistency, the bounds on the various components are estimated at a certain time frame based on the bounds of the components at the previous time frame that spatially intersect with it.

Although our own method shares many features with these techniques, it differs in two important respects that we will highlight: First, we combine the usual color and motion models with a sophisticated approach based on a generative model to estimating the probabilities of occupancy, which explicitly handles complex occlusion interactions between detected individuals, as will be discussed in Section 5. Second, we rely on dynamic programming to ensure greater stability in challenging situations by simultaneously handling multiple frames.

3 Problem Formulation

Our goal is to track an a priori unknown number of people from a few synchronized video streams taken at head level. In this section, we formulate this problem as one of finding the most probable state of a hidden Markov process, given the set of images acquired at each time step, which we will refer to as a *temporal frame*. We then briefly outline the computation of the relevant probabilities by using the notations summarized in Tables 1 and 2, which we also use in the following two sections to discuss in more details the actual computation of those probabilities.

3.1 Computing the Optimal Trajectories

We process the video sequences by batches of $T = 100$ frames, each of which includes C images, and we compute the most likely trajectory for each individual. To achieve consistency over successive batches, we only keep the result on the first 10 frames and slide our temporal window. This is illustrated in Fig. 3.

We discretize the visible part of the ground plane into a finite number G of regularly spaced 2D locations and we introduce a virtual hidden location $\mathcal{H}$ that will be used to model entrances and departures from and into the visible area. For a given batch, let $\mathbf{L}_t = (L_t^1, \ldots, L_t^{N^*})$ be the hidden stochastic processes standing for the locations of individuals, whether visible or not. The number N^* stands for the maximum allowable number of individuals in our world. It is large enough so that conditioning on the number of visible ones does not change the probability of a new individual entering the scene. The L_t^n variables therefore take values in $\{1, \ldots, G, \mathcal{H}\}$.

Given $\mathbf{I}_t = (I_t^1, \ldots, I_t^C)$, the images acquired at time t for $1 \leq t \leq T$, our task is to find the values of $\mathbf{L}_1, \ldots, \mathbf{L}_T$ that maximize

$$P(\mathbf{L}_1, \ldots, \mathbf{L}_T \,|\, \mathbf{I}_1, \ldots, \mathbf{I}_T). \tag{1}$$

As will be discussed in Section 4.1, we compute this maximum a posteriori in a greedy way, processing one individual at a time, including the hidden ones who can move into the visible scene or not. For each one, the algorithm performs the computation, under the constraint that no individual can be at a visible location occupied by an individual already processed.

In theory, this approach could lead to undesirable local minima, for example, by connecting the trajectories of two separate people. However, this does not happen often because our batches are sufficiently long. To further reduce the chances of this, we process individual trajectories in an order that depends on a reliability score so that the most reliable ones are computed first, thereby reducing the potential for confusion when processing the remaining ones. This order also ensures that if an individual remains in the hidden location, then all the other people present in the hidden location will also stay there and, therefore, do not need to be processed.

TABLE 1
Notations (Deterministic Quantities)

$W \times H$	image resolution.
C	number of cameras.
G	number of locations in the ground discretization ($\simeq 1000$).
T	number of frames processed in one batch ($= 100$).
t	frame index.
$I \otimes J$	intersection of images, $\forall (x,y),\ (I \otimes J)(x,y) = I(x,y)J(x,y)$.
$I \oplus J$	disjunction of images, $\forall (x,y),\ (I \oplus J)(x,y) = 1 - (1 - I(x,y))(1 - J(x,y))$.
Ψ	a pseudo-distance between images.
Q	the product law used to approximate, for a fixed t, the real posterior distribution $P(\cdot \mid \mathbf{B}_t)$.
E_Q	Expectation under $\mathbf{X} \sim Q$.
q_k	the marginal probability of Q, that is $Q(X_k = 1)$.
ϵ_k	the prior probability of presence at location i, $P(X_k = 1)$.
λ_k	is $\log \frac{1-\epsilon_k}{\epsilon_k}$, the log-ratio of the prior probability.
$\mathcal{A}_k^c$	the image composed of 1s inside a rectangle standing for the silhouette of an individual at location k seen from camera c, and 0s elsewhere.
N^*	virtual number of people, including the non-visible ones.
μ_n^c	color distribution of individual n from camera c.

TABLE 2
Notations (Random Quantities)

$\mathbf{I}_t$	images from all the cameras $\mathbf{I}_t = (I_t^1, \ldots, I_t^C)$.
$\mathbf{B}_t$	binary images generated by the background subtraction $\mathbf{B}_t = (B_t^1, \ldots, B_t^C)$.
$\mathbf{T}_t$	texture information.
A_t^c	ideal random image generated by putting rectangles $\mathcal{A}_k^c$ where $X_t^k = 1$, thus a function of $\mathbf{X}_t$.
$\overline{A}_{k,\xi}^c$	compact notation for the average synthetic image $E_Q(A^c \mid X_k = \xi)$, see Figure 6.
$\mathbf{L}_t$	vector of people locations on the ground plane or in the hidden location $\mathbf{L}_t = (L_t^1, \ldots, L_t^{N^*})$. Each of these random variables takes values into $\{1, \ldots, G, \mathcal{H}\}$, where $\mathcal{H}$ is the hidden place.
$\mathbf{L}^n$	trajectory of individual n, $\mathbf{L}^n = (L_1^n, \ldots, L_T^n)$.
$\mathbf{X}_t$	vectors of boolean random variable $(X_t^1, \ldots, X_t^G)$ standing for the occupancy of location k on the ground plane $\left(X_t^k = 1\right) \Leftrightarrow (\exists n, L_t^n = k)$.

Our experimental results show that our method does not suffer from the usual weaknesses of greedy algorithms such as a tendency to get caught in bad local minima. We therefore believe that it compares very favorably to stochastic optimization techniques in general and more specifically particle filtering, which usually requires careful tuning of metaparameters.

Fig. 3. Video sequences are processed by batch of 100 frames. Only the first 10 percent of the optimization result is kept and the rest is discarded. The temporal window is then slid forward and the optimization is repeated on the new window.

3.2 Stochastic Modeling

We will show in Section 4.2 that since we process individual trajectories, the whole approach only requires us to define a valid motion model $P(L_{t+1}^n | L_t^n = k)$ and a sound appearance model $P(\mathbf{I}_t | L_t^n = k)$.

The motion model $P(L_{t+1}^n | L_t^n = k)$, which will be introduced in Section 4.3, is a distribution into a disc of limited radius and center k, which corresponds to a loose bound on the maximum speed of a walking human. Entrance into the scene and departure from it are naturally modeled, thanks to the

hidden location $\mathcal{H}$, for which we extend the motion model. The probabilities to enter and to leave are similar to the transition probabilities between different ground plane locations.

In Section 4.4, we will show that the appearance model $P(\mathbf{I}_t|L_t^n = k)$ can be decomposed into two terms. The first, described in Section 4.5, is a very generic color-histogram-based model for each individual. The second, described in Section 5, approximates the marginal conditional probabilities of occupancy of the ground plane, given the results of a background subtraction algorithm, in all views acquired at the same time. This approximation is obtained by minimizing the Kullback-Leibler divergence between a product law and the true posterior. We show that this is equivalent to computing the marginal probabilities of occupancy so that under the product law, the images obtained by putting rectangles of human sizes at occupied locations are likely to be similar to the images actually produced by the background subtraction.

This represents a departure from more classical approaches to estimating probabilities of occupancy that rely on computing a visual hull [26]. Such approaches tend to be pessimistic and do not exploit trade-offs between the presence of people at different locations. For instance, if due to noise in one camera, a person is not seen in a particular view, then he would be discarded, even if he were seen in all others. By contrast, in our probabilistic framework, sufficient evidence might be present to detect him. Similarly, the presence of someone at a specific location creates an occlusion that hides the presence behind, which is not accounted for by the hull techniques but is by our approach.

Since these marginal probabilities are computed independently at each time step, they say nothing about identity or correspondence with past frames. The appearance similarity is entirely conveyed by the color histograms, which has experimentally proved sufficient for our purposes.

4 Computation of the Trajectories

In Section 4.1, we break the global optimization of several people's trajectories into the estimation of optimal individual trajectories. In Section 4.2, we show how this can be performed using the classical Viterbi's algorithm based on dynamic programming. This requires a motion model given in Section 4.3 and an appearance model described in Section 4.4, which combines a color model given in Section 4.5 and a sophisticated estimation of the ground plane occupancy detailed in Section 5.

We partition the visible area into a regular grid of G locations, as shown in Figs. 5c and 6, and from the camera calibration, we define for each camera c a family of rectangular shapes $\mathcal{A}_1^c, \ldots, \mathcal{A}_G^c$, which correspond to crude human silhouettes of height 175 cm and width 50 cm located at every position on the grid.

4.1 Multiple Trajectories

Recall that we denote by $\mathbf{L}^n = (L_1^n, \ldots, L_T^n)$ the trajectory of individual n. Given a batch of T temporal frames $\mathbf{I} = (\mathbf{I}_1, \ldots, \mathbf{I}_T)$, we want to maximize the posterior conditional probability:

$$P(\mathbf{L}^1 = \mathbf{l}^1, \ldots, \mathbf{L}^{N^*} = \mathbf{l}^{N^*}|\mathbf{I}) = P(\mathbf{L}^1 = \mathbf{l}^1|\mathbf{I}) \prod_{n=2}^{N^*} P\big(\mathbf{L}^n = \mathbf{l}^n|\mathbf{I}, \mathbf{L}^1 = \mathbf{l}^1, \ldots, \mathbf{L}^{n-1} = \mathbf{l}^{n-1}\big). \quad (2)$$

Simultaneous optimization of all the L^is would be intractable. Instead, we optimize one trajectory after the other, which amounts to looking for

$$\hat{\mathbf{l}}^1 = \arg\max_l P(\mathbf{L}^1 = l|\mathbf{I}), \quad (3)$$

$$\hat{\mathbf{l}}^2 = \arg\max_l P(\mathbf{L}^2 = l|\mathbf{I}, \mathbf{L}^1 = \hat{\mathbf{l}}^1), \quad (4)$$

$$\vdots$$

$$\hat{\mathbf{l}}^{N^*} = \arg\max_l P(\mathbf{L}^{N^*} = l|\mathbf{I}, \mathbf{L}^1 = \hat{\mathbf{l}}^1, \mathbf{L}^2 = \hat{\mathbf{l}}^2, \ldots). \quad (5)$$

Note that under our model, conditioning one trajectory, given other ones, simply means that it will go through no already occupied location. In other words,

$$P(\mathbf{L}^n = l|\mathbf{I}, \mathbf{L}^1 = \hat{\mathbf{l}}^1, \ldots, \mathbf{L}^{n-1} = \hat{\mathbf{l}}^{n-1}) = P(\mathbf{L}^n = l|\mathbf{I}, \forall k < n, \forall t, L_t^n \neq \hat{l}_t^k), \quad (6)$$

which is $P(\mathbf{L}^n = l|\mathbf{I})$ with a reduced set of the admissible grid locations.

Such a procedure is recursively correct: If all trajectories estimated up to step n are correct, then the conditioning only improves the estimate of the optimal remaining trajectories. This would suffice if the image data were informative enough so that locations could be unambiguously associated to individuals. In practice, this is obviously rarely the case. Therefore, this greedy approach to optimization has undesired side effects. For example, due to partly missing localization information for a given trajectory, the algorithm might mistakenly start following another person's trajectory. This is especially likely to happen if the tracked individuals are located close to each other.

To avoid this kind of failure, we process the images by batches of $T = 100$ and first extend the trajectories that have been found with high confidence, as defined below, in the previous batches. We then process the lower confidence ones. As a result, a trajectory that was problematic in the past and is likely to be problematic in the current batch will be optimized last and, thus, prevented from "stealing" somebody else's location. Furthermore, this approach increases the spatial constraints on such a trajectory when we finally get around to estimating it.

We use as a confidence score the concordance of the estimated trajectories in the previous batches and the localization cue provided by the estimation of the probabilistic occupancy map (POM) described in Section 5. More precisely, the score is the number of time frames where the estimated trajectory passes through a local maximum of the estimated probability of occupancy. When the POM does not detect a person on a few frames, the score will naturally decrease, indicating a deterioration of the localization information. Since there is a high degree of overlapping between successive batches, the challenging segment of a trajectory, which is due to the failure of the background subtraction or change in illumination, for instance, is met in several batches before it actually happens during the 10 kept frames. Thus, the heuristic would have ranked the corresponding individual in the last ones to be processed when such problem occurs.

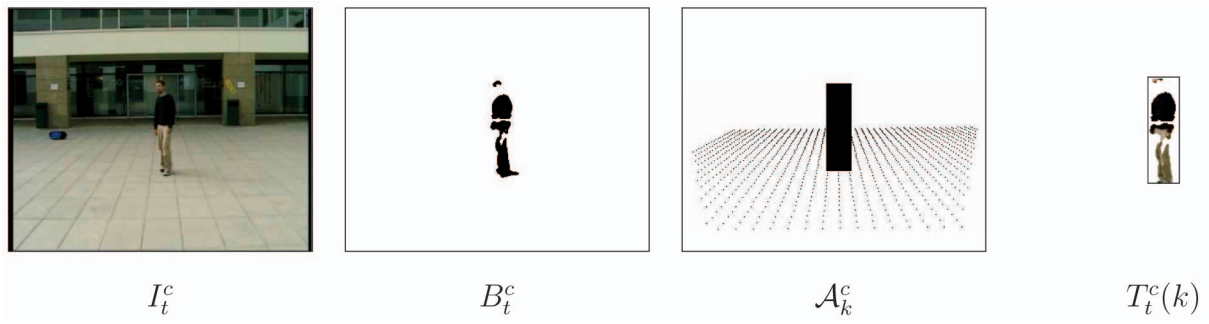

Fig. 4. The color model relies on a stochastic modeling of the color of the pixels $T_t^c(k)$ sampled in the intersection of the binary image B_t^c produced by the background subtraction and the rectangle $\mathcal{A}_k^c$ corresponding to the location k.

4.2 Single Trajectory

Let us now consider only the trajectory $\mathbf{L}^n = (L_1^n, \ldots, L_T^n)$ of individual n over T temporal frames. We are looking for the values $(l_1^n, \ldots, l_T^n)$ in the subset of free locations of $\{1, \ldots, G, \mathcal{H}\}$. The initial location l_1^n is either a known visible location if the individual is visible in the first frame of the batch or $\mathcal{H}$ if he is not. We therefore seek to maximize

$$\begin{aligned} &P(L_1^n = l_1^n, \ldots, L_T^n = l_t^n | \mathbf{I}_1, \ldots, \mathbf{I}_T) \\ &\quad = \frac{P(\mathbf{I}_1, L_1^n = l_1^n, \ldots, \mathbf{I}_T, L_T^n = l_T^n)}{P(\mathbf{I}_1, \ldots, \mathbf{I}_T)}. \end{aligned} \tag{7}$$

Since the denominator is constant with respect to $\mathbf{l}^n$, we simply maximize the numerator, that is, the probability of both the trajectories and the images. Let us introduce the maximum of the probability of both the observations and the trajectory ending up at location k at time t:

$$\Phi_t(k) = \max_{l_1^n, \ldots, l_{t-1}^n} P(\mathbf{I}_1, L_1^n = l_1^n, \ldots, \mathbf{I}_t, L_t^n = k). \tag{8}$$

We model jointly the processes L_t^n and $\mathbf{I}_t$ with a hidden Markov model, that is

$$P(L_{t+1}^n | L_t^n, L_{t-1}^n, \ldots) = P(L_{t+1}^n | L_t^n) \tag{9}$$

and

$$P(I_t, I_{t-1}, \ldots | L_t^n, L_{t-1}^n, \ldots) = \prod_t P(I_t | L_t^n). \tag{10}$$

Under such a model, we have the classical recursive expression

$$\Phi_t(k) = \underbrace{P(\mathbf{I}_t | L_t^n = k)}_{\text{Appearance model}} \max_\tau \underbrace{P(L_t^n = k | L_{t-1}^n = \tau)}_{\text{Motion model}} \Phi_{t-1}(\tau) \tag{11}$$

to perform a global search with dynamic programming, which yields the classic Viterbi algorithm. This is straightforward, since the L_t^ns are in a finite set of cardinality $G + 1$.

4.3 Motion Model

We chose a very simple and unconstrained motion model:

$$P(L_t^n = k | L_{t-1}^n = \tau) = \begin{cases} 1/Z \cdot e^{-\rho \|k - \tau\|} & \text{if } \|k - \tau\| \leq c \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where the constant ρ tunes the average human walking speed, and c limits the maximum allowable speed. This probability is isotropic, decreases with the distance from location k, and is zero for $\|k - \tau\|$ greater than a constant maximum distance. We use a very loose maximum distance c of one square of the grid per frame, which corresponds to a speed of almost 12 mph. We also define explicitly the probabilities of transitions to the parts of the scene that are connected to the hidden location $\mathcal{H}$. This is a single door in the indoor sequences and all the contours of the visible area in the outdoor sequences in Fig. 1. Thus, entrance and departure of individuals are taken care of naturally by the estimation of the maximum a posteriori trajectories. If there are enough evidence from the images that somebody enters or leaves the room, then this procedure will estimate that the optimal trajectory does so, and a person will be added to or removed from the visible area.

4.4 Appearance Model

From the input images $\mathbf{I}_t$, we use background subtraction to produce binary masks $\mathbf{B}_t$ such as those in Fig. 4. We denote as $\mathbf{T}_t$ the colors of the pixels inside the blobs and treat the rest of the images as background, which is ignored.

Let X_k^t be a Boolean random variable standing for the presence of an individual at location k of the grid at time t. In Appendix B, we show that

$$\overbrace{P(\mathbf{I}_t | L_t^n = k)}^{\text{Appearance model}} \propto \underbrace{P(L_t^n = k | X_t^k = 1, \mathbf{T}_t)}_{\text{Color model}} \underbrace{P(X_t^k = 1 | \mathbf{B}_t)}_{\text{Ground plane occupancy}}. \tag{13}$$

The ground plane occupancy term will be discussed in Section 5, and the color model term is computed as follows.

4.5 Color Model

We assume that if someone is present at a certain location k, then his presence influences the color of the pixels located at the intersection of the moving blobs and the rectangle $\mathcal{A}_k^c$ corresponding to the location k. We model that dependency as if the pixels were independent and identically distributed and followed a density in the red, green, and blue (RGB) space associated to the individual. This is far simpler than the color models used in either [18] or [13], which split the body area in several subparts with dedicated color distributions, but has proved sufficient in practice.

If an individual n was present in the frames preceding the current batch, then we have an estimation for any camera c of his color distribution μ_n^c, since we have previously collected the pixels in all frames at the locations

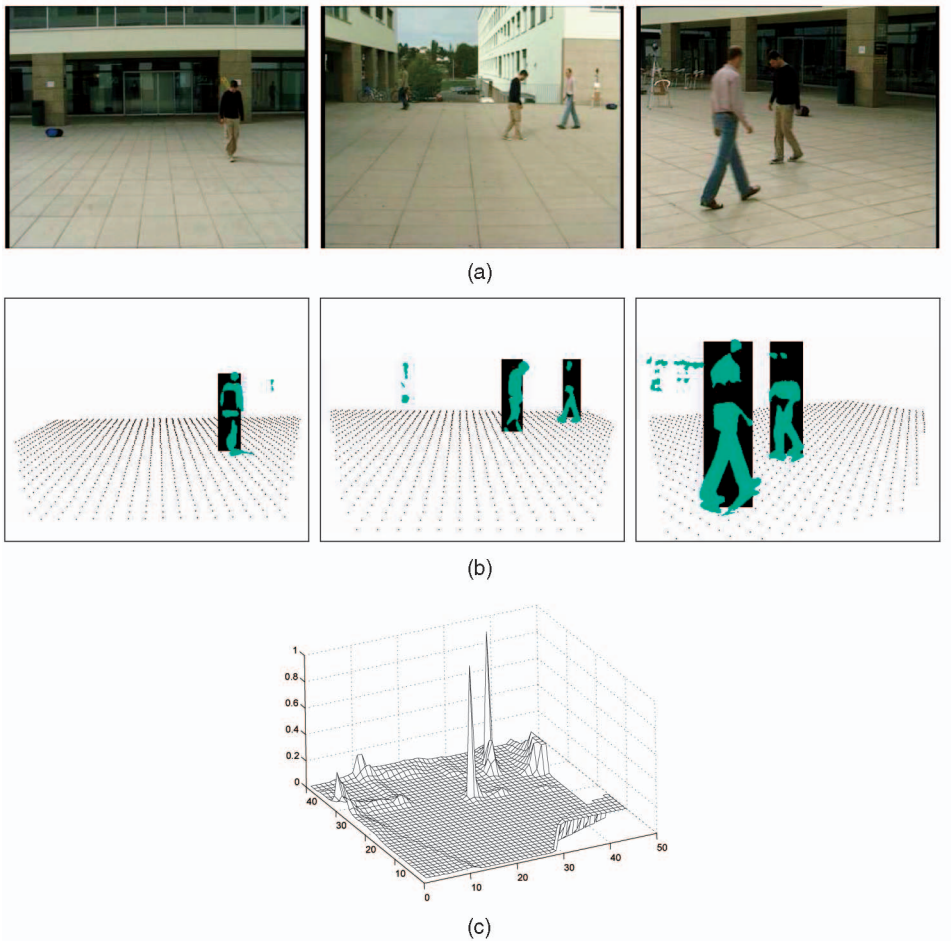

Fig. 5. (a) Original images from three cameras. (b) Binary blobs produced by background subtraction and synthetic average images computed from them by the estimation of the POM algorithm. (c) The graph represents the corresponding occupancy probabilities q_k on the grid.

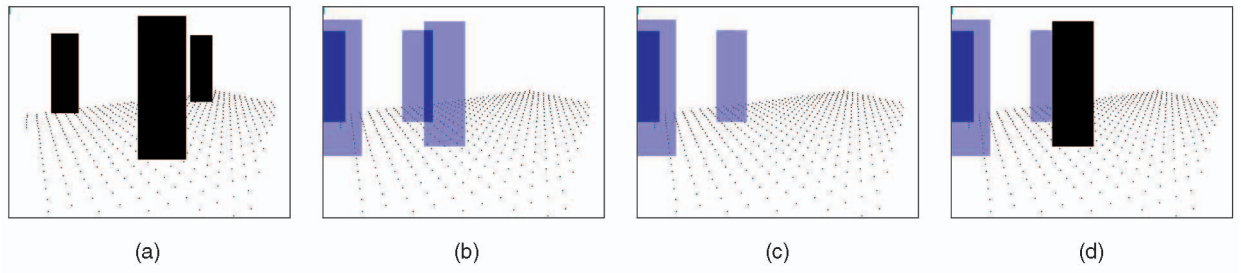

Fig. 6. (a) A synthetic picture A^c with three X^ks equal to 1. (b) The average image $E_Q(A^c)$, where all q_ks are null, but four of them are equal to 0.2. (c) and (d) $\overline{A}^c_{k,0} = E_Q(A^c|X^k = 0)$ and $\overline{A}^c_{k,1} = E_Q(A^c|X^k = 1)$, respectively, where k is the location corresponding to the black rectangle in (d).

of his estimated trajectory. If he is at the hidden location $\mathcal{H}$, then we consider that his color distribution μ^c_n is flat.

Let $T^c_t(k)$ denote the pixels taken at the intersection of the binary image produced by the background subtraction from the stream of camera c at time t and the rectangle $\mathcal{A}^c_k$ corresponding to location k in that same field of view (see Fig. 4). Note that even if an individual is actually at that location, this intersection can be empty if the background subtraction fails.

Let $\mu^c_1, \ldots, \mu^c_{N^*}$ be the color distributions of the N^* individuals present in the scene at the beginning of the batch of T frames that we are processing for camera c. The distribution may vary with the camera due to the difference in the camera technology or illumination angle. We have (see Appendix C)

$$\overbrace{P(L^n_t = k|X^k_t = 1, \mathbf{T}_t)}^{\text{Color model}} = \frac{P(\mathbf{T}_t|L^n_t = k)}{\sum_m P(\mathbf{T}_t|L^m_t = k)}, \tag{14}$$

where

$$P(\mathbf{T}_t|L^n_t = k) = P(T^1_t(k), \ldots, T^C_t(k)|L^n_t = k), \tag{15}$$

$$= \prod_{c=1}^{C} \prod_{r \in T^c_t(k)} \mu^c_n(r). \tag{16}$$

5 Probabilistic Occupancy Map

The algorithm described in the previous section requires estimates at every location of the probability that somebody is standing there, given the evidence provided by the

background subtraction. These probabilities are written as $P(X_t^k = 1|\mathbf{B}_t)$ for every k and t and appear in (13).

As discussed in Section 3.2, to estimate accurately the probabilities of presence at every location, we need to take into account both the information provided in each separate view and the couplings between views produced by occlusions. Instead of combining heuristics related to geometrical consistency or sensor noise, we encompass all the available prior information that we have about the task in a generative model of the result of the background subtraction, given the true state of occupancy $(X_t^1, \ldots, X_t^G)$ that we are trying to estimate.

Ideally, provided with such a model of $P(\mathbf{B}_t|\mathbf{X}_t)$, that is, of the result of the background subtraction, given the true state of occupancy of the scene, estimating $P(\mathbf{X}_t|\mathbf{B}_t)$ becomes a Bayesian computation. However, due to the complexity of any nontrivial model of $P(\mathbf{B}_t|\mathbf{X}_t)$ and to the dimensionality of both $\mathbf{B}_t$ and $\mathbf{X}_t$, this cannot be done with a generic method.

To address this problem, we represent humans as simple rectangles and use them to create synthetic ideal images that we would observe if people were at given locations. Under this model of the image, given the true state, we approximate the occupancy probabilities as the marginals of a product law Q minimizing the Kullback-Leibler divergence from the "true" conditional posterior distribution. This allows us to compute these probabilities as the fixed point of a large system of equations, as discussed in detail in this section.

More specifically, in Section 5.1, we introduce two independence assumptions, under which we derive the analytical results of the other sections, and argue that they are legitimate. In Section 5.2, we propose our generative model of $P(\mathbf{B}|\mathbf{X})$, which involves measuring the distance between the actual images $\mathbf{B}$ and a crude synthetic image that is a function of the $\mathbf{X}$. From these assumptions and model, we derive in Section 5.3 an analytical relation between estimates $q_1, \ldots, q_G$ of the marginal probabilities of occupancy $P(X_t^1 = 1|\mathbf{B}_t), \ldots, P(X_t^G = 1|\mathbf{B}_t)$ by minimizing the Kullback-Leibler divergence between the corresponding product law and the true posterior. This leads to a fast iterative algorithm that estimates them as the solution of a fixed-point problem, as shown in Section 5.4.

Since we do this at each time step separately, we drop t from all notations in the remainder of this section for clarity.

5.1 Independence Assumptions

We introduce here two assumptions of independence that will allow us to derive analytically the relation between the optimal q_ks.

Our first assumption is that individuals in the room do not take into account the presence of other individuals in their vicinity when moving around, which is true, as long as avoidance strategies are ignored. This can be formalized as

$$P(X^1, \ldots, X^G) = \prod_k P(X^k). \tag{17}$$

Our second assumption involves considering that all statistical dependencies between views are due to the presence of individuals in the room. This is equivalent to treating the views as functions of the vector $\mathbf{X} = (X^1, \ldots, X^G)$ plus some independent noise. This implies that as soon as the presence of all individuals is known, the views become independent. This is true, as long as we ignore other hidden variables such as morphology or garments that may simultaneously influence several views. This assumption can be written down as

$$P(B^1, \ldots, B^C|\mathbf{X}) = \prod_c P(B^c|\mathbf{X}). \tag{18}$$

5.2 Generative Image Model

To relate the values of the X^ks to the images produced by background subtraction $B^1, \ldots, B^C$, we propose here a model of the latter given the former.

Let A^c be the synthetic image obtained by putting rectangles at locations where $X^k = 1$ (see an example in Fig. 6a). Thus, $A^c = \oplus_k X^k \mathcal{A}_k^c$, where $\oplus$ denotes the "union" between two images. Such an image is a function of $\mathbf{X}$ and, thus, a random quantity. We model the image B^c produced by the background subtraction as if it was this ideal image with some random noise.

As it appears empirically that the noise increases with the area of the ideal image A^c, we introduce a normalized pseudodistance Ψ to account for this asymmetry. For any gray-scale image $A \in [0,1]^{W \times H}$, we denote by $|A|$ the sum of its pixels, and we denote by $\otimes$ the product per pixel of two images. We introduce Ψ defined by

$$\forall B, A \in [0,1]^{W \times H}, \quad \Psi(B, A) = \frac{1}{\sigma} \frac{|B \otimes (1 - A) + (1 - B) \otimes A|}{|A|}. \tag{19}$$

Also, we model the conditional distribution $P(B^c|\mathbf{X})$ of the background subtraction images, given the true hidden state, as a density decreasing with the pseudodistance $\Psi(B^c, A^c)$ between the image produced by the background subtraction and an image A^c obtained by putting rectangular shapes, where people are present according to $\mathbf{X}$. We end up with the following model:

$$P(\mathbf{B}|\mathbf{X}) = \prod_c P(B^c|\mathbf{X}), \tag{20}$$

$$= \prod_c P(B^c|A^c), \tag{21}$$

$$= \frac{1}{Z} \prod_c e^{-\Psi(B^c, A^c)}. \tag{22}$$

The parameter σ accounts for the quality of the background subtraction. The smaller the σ is, the more the B^c is picked around its ideal value A^c. The value of σ was fixed arbitrarily to 0.01, but experiments demonstrated that the algorithm is not sensitive to that value.

5.3 Relation between the q_ks

We denote by E_Q the expectation under $\mathbf{X} \sim Q$. Since we want to minimize the Kullback-Leibler divergence between the approximation Q and the "true" posterior $P(\cdot|\mathbf{B})$, we use the following form of its derivative with respect to the unknown q_k (see Appendix A):

$$\frac{\partial}{\partial q_k} KL(Q, P(\cdot|\mathbf{B})) = \log \frac{q_k(1 - \epsilon_k)}{(1 - q_k)\epsilon_k} + E_Q\left(\sum_c \Psi(B^c, A^c) \middle| X^k = 1\right) - E_Q\left(\sum_c \Psi(B^c, A^c) \middle| X^k = 0\right). \tag{23}$$

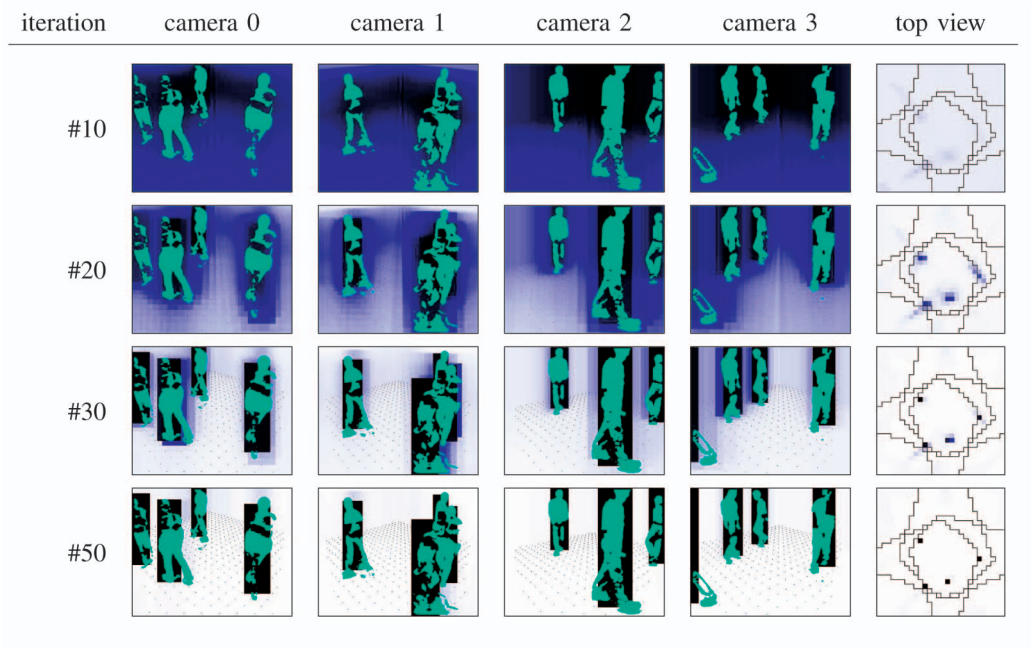

Fig. 7. Convergence process for the estimation of the POM. Camera views show both the background subtraction blobs and the synthetic average image corresponding to different iterations.

Hence, if we solve

$$\frac{\partial}{\partial q_k} KL(Q, P(\cdot|\mathbf{B})) = 0, \tag{24}$$

then we obtain

$$q_k = \frac{1}{1+\exp\left(\lambda_k + \sum_c E_Q(\Psi(B^c, A^c)|X^k=1) - E_Q(\Psi(B^c, A^c)|\mathbf{X})|X^k=0)\right)} \tag{25}$$

with $\lambda_k = \log\frac{1-\epsilon_k}{\epsilon_k}$.

Unfortunately, the computation of $E_Q(\Psi(B^c, A^c)|X^k = \xi)$ is untractable. However, since under $\mathbf{X} \sim Q$, the image A^c is concentrated around B^c, we approximate the following $\forall \xi \in \{0, 1\}$:

$$E_Q(\Psi(B^c, A^c)|X^k = \xi) \simeq \Psi(B^c, E_Q(A^c|X^k = \xi)). \tag{26}$$

This leads to our main result:

$$q_k = \frac{1}{1+\exp\left(\lambda_k + \sum_c \Psi(B^c, E_Q(A^c|X^k=1)) - \Psi(B^c, E_Q(A^c|X^k=0))\right)}. \tag{27}$$

Note that the conditional synthetic images $E_Q(A^c|X^k = 0)$ and $E_Q(A^c|X^k = 1)$ are equal to $E_Q(A^c)$, with q_k forced to 0 or 1, respectively, as shown in Fig. 6. Since Q is a product law, we have the following for any pixel (x, y):

$$E_Q(A^c(x, y)) = Q(A^c(x, y) = 1), \tag{28}$$

$$= 1 - Q\Big(\forall k, \mathcal{A}_k^c(x, y)X^k = 0\Big), \tag{29}$$

$$= 1 - \prod_{k:\mathcal{A}_k^c(x,y)=1} (1 - q_k). \tag{30}$$

Finally, $E_Q(A^c|X^k = \xi)$ are functions of the $(q_l)_{l\neq k}$ and (27) can be seen as one equation of a large system whose unknowns are the q_ks. Fig. 7 shows the evolution of both the marginals q_k and the average images $E_Q(A^c)$ during the iterative estimation of the solution.

Intuitively, if putting the rectangular shape for position k in the image improves the fit with the actual images, the score $\Psi(B^c, E_Q(A^c|X^k = 1))$ decreases, $\Psi(B^c, E_Q(A^c|X^k = 0))$ increases, and the sum in the exponential is negative, leading to a larger q_k. Note that occlusion is taken into account naturally: if a rectangular shape at position k is occluded by another one whose presence is very likely, then the value of q_k does not influence the average image, and all terms vanish, except for λ_k in the exponential. Thus, the resulting q_k remains equal to the prior.

5.4 Fast Estimation of the q_ks

We estimate the q_ks as follows: We first give them a uniform value and use them to compute the average synthetic images $\overline{A}^c_{k,\xi} = E_Q(A^c|X^k = \xi)$. We then re-estimate every q_k with (27) and iterate the process until a stable solution is reached.

The main remaining issue is the computation of $\Psi(B^c, \overline{A}^c_{k,\xi})$, which has to be done G times per iteration for as many iterations as required to converge, which is usually of the order of 100.

Fortunately, the images $E_Q(A^c)$ and $\overline{A}^c_{k,\xi}$ differ only in the rectangle $\mathcal{A}_k$, where the latter is multiplied by a constant factor. Hence, we can show that by using integral images, we can compute the distance from the true image produced by the background subtraction to the average image obtained with one of the q_ks modified at constant time and very rapidly.

We organize the computation to take advantage of that trick and finally perform the following steps at each iteration of our algorithm.

Let $\oplus$ denote the pixelwise disjunction operator between binary images (the "union" image), $\otimes$ be the pixelwise product (the "intersection" image), $|I|$ be the sum of the pixels of an image I, and let 1 be the constant image whose pixels are all equal to 1:

$$\overline{A}^c = 1 - \otimes_k \left(1 - q_k \mathcal{A}_k^c\right), \tag{31}$$

$$\left|\overline{A}_{k,\xi}^c\right| = \left|\overline{A}^c\right| + \frac{\xi - q_k}{1 - q_k}\left|\left(1 - \overline{A}^c\right) \otimes \mathcal{A}_k^c\right|, \tag{32}$$

$$\left|B_c \otimes \overline{A}_{k,\xi}^c\right| = \left|B_c \otimes \overline{A}^c\right| + \frac{\xi - q_k}{1 - q_k}\left|B_c \otimes \left(1 - \overline{A}^c\right) \otimes \mathcal{A}_k^c\right|, \tag{33}$$

$$\Psi(B_c, \overline{A}_{k,\xi}^c) = \frac{1}{\sigma}\frac{|B_c| - 2|B_c \otimes \overline{A}_{k,\xi}^c| + |\overline{A}_{k,\xi}^c|}{|\overline{A}_{k,\xi}^c|}, \tag{34}$$

$$q_k \leftarrow \frac{1}{1 + \exp\left(\lambda_k + \sum_c \Psi(B_c, \overline{A}_{k,1}^c) - \Psi(B_c, \overline{A}_{k,0}^c)\right)}. \tag{35}$$

At each iteration and for every c, Step (31) involves computing the average of the synthetic image under Q with the current estimates of q_ks. Steps (32) and (33), respectively, sum the pixels of the conditional average images, given X^k, and of the same image multiplied pixelwise by the output of the background subtraction. This is done at the same time for every k and uses precomputed integral images of $1 - \overline{A}^c$ and $B_c \otimes (1 - \overline{A}^c)$, respectively. Finally, Steps (34) and (35) return the distance between the result of the background subtraction and the conditional average synthetic under Q, as well as the corresponding updated marginal probability. Except for the exponential in the last step, which has to be repeated at every location, the computation only involves sums and products and is therefore fast.

6 Results

In our implementation, we first compute the POM described in Section 5 separately at each time step and then use the results as input to the dynamic programming approach of Section 4. We describe first the sequences used for the experiments and the background subtraction algorithm. Then, we present the results obtained with the estimation of the POM in individual time frames. Finally, we present the result of our global optimization.

6.1 Video Sequences

We use here two indoor and four outdoor sequences that were shot on our campus. The frame rate for all of the sequences is 25 fps.

6.1.1 Indoor Sequences

The two sequences depicted in the upper row of Fig. 1 and the two upper rows of Fig. 9 were shot by a video-surveillance-dedicated setup of four synchronized cameras in a $50\ \mathrm{m}^2$ room. Two cameras were roughly at head level ($\simeq$ 1.80 m), and the two others were slightly higher ($\simeq$ 2.30 m). They were located at each corner of the room. The first sequence is 3,800 frames long and shows four individuals entering the room and walking continuously. The second contains 2,800 frames and involves six individuals. This actually results in a more complex reconstruction problem than usually happens in real-life situations mostly because people tend to occlude each other much more often.

In these sequences, the area of interest was of size $5.5\ \mathrm{m} \times 5.5\ \mathrm{m} \simeq 30\ \mathrm{m}^2$ and discretized into $G = 28 \times 28 = 794$ locations, corresponding to a regular grid with a 20 cm resolution.

6.1.2 Outdoor Sequences

The outdoor sequences shown in the lower row of Fig. 1 and the four lower rows of Fig. 9 were shot at two different locations on our campus. We used respectively three and four standard and unsynchronized digital video cameras and synchronized the video streams by hand afterward. All cameras were at head level ($\simeq$ 1.80 m), covering the area of interest from different angles. The ground is flat, with a regular pavement at both locations.

The area of interest at the first location is of size 10 m $\times$ 10 m and discretized into $G = 40 \times 40 = 1,600$ locations, corresponding to a regular grid with a resolution of 25 cm. Up to four individuals appear simultaneously. At the second location, the area is of size 6 m $\times$ 10 m and discretized into $G = 30 \times 50 = 1,500$ locations. Additionally, tables and chairs have been placed at the center to simulate static obstacles. The sequences were shot early in the morning, and the sun, which is low on the horizon, produces very long and sharp shadows. Up to six people appear on those videos, most of which are dressed in very similar colors.

6.2 Background Subtraction

For the generation of the foreground blobs, we have used two different background subtraction algorithms. Most of our test sequences have been processed by a background subtraction program developed at *Visiowave* [27], which generates nicely smoothed blobs (see Figs. 5 and 7, for example). For outdoor sequences with severe illumination changes, we used our own implementation of a background subtraction algorithm using eigenbackgrounds [22].

6.3 POM at Individual Time Steps

Fig. 7 displays the convergence of the POM estimation alone on a single time frame, whereas Fig. 8 shows its detection results on both the indoor and outdoor sequences. These results are only about *detection*. The algorithm operated on a frame-by-frame basis, with no time consistency and, thus, does not maintain identities of the people across frames.

As can be seen from the screen shots, the accuracy of the POM is generally very good. Its performance is, however, correlated with the quality of the background subtraction blobs. An inaccurate blob detection, as shown in frame 2643 in Fig. 8, can lead to incorrect people detection. As the video sequences demonstrate, POM remains robust with silhouettes as small as 70 pixels in height, which roughly corresponds to a distance of 15 m when using a standard focal.

On one of the indoor sequences, we have computed precise error rates by counting in each frame the number of actually present individuals, the number of detected individuals, and the number of false detections. We defined a correct detection as one for which the reprojected boxes intersect the individual on at least three camera views out of four. Such a tolerance accommodates for cases where due to optical deformations, although the estimated location is correct, the reprojection does not match the real location on one view. With such a definition, the estimated false-negative error rate on the indoor video sequence with four people is 6.14 percent, and the false-positive error rate is 3.99 percent. In absolute terms, our detection performance is excellent, considering that we use only a small proportion of the available image information.

6.4 Global Trajectory Estimation

Since the observed area is discretized into a finite number of positions, we improve the result accuracy by linearly interpolating the trajectories on the output images.

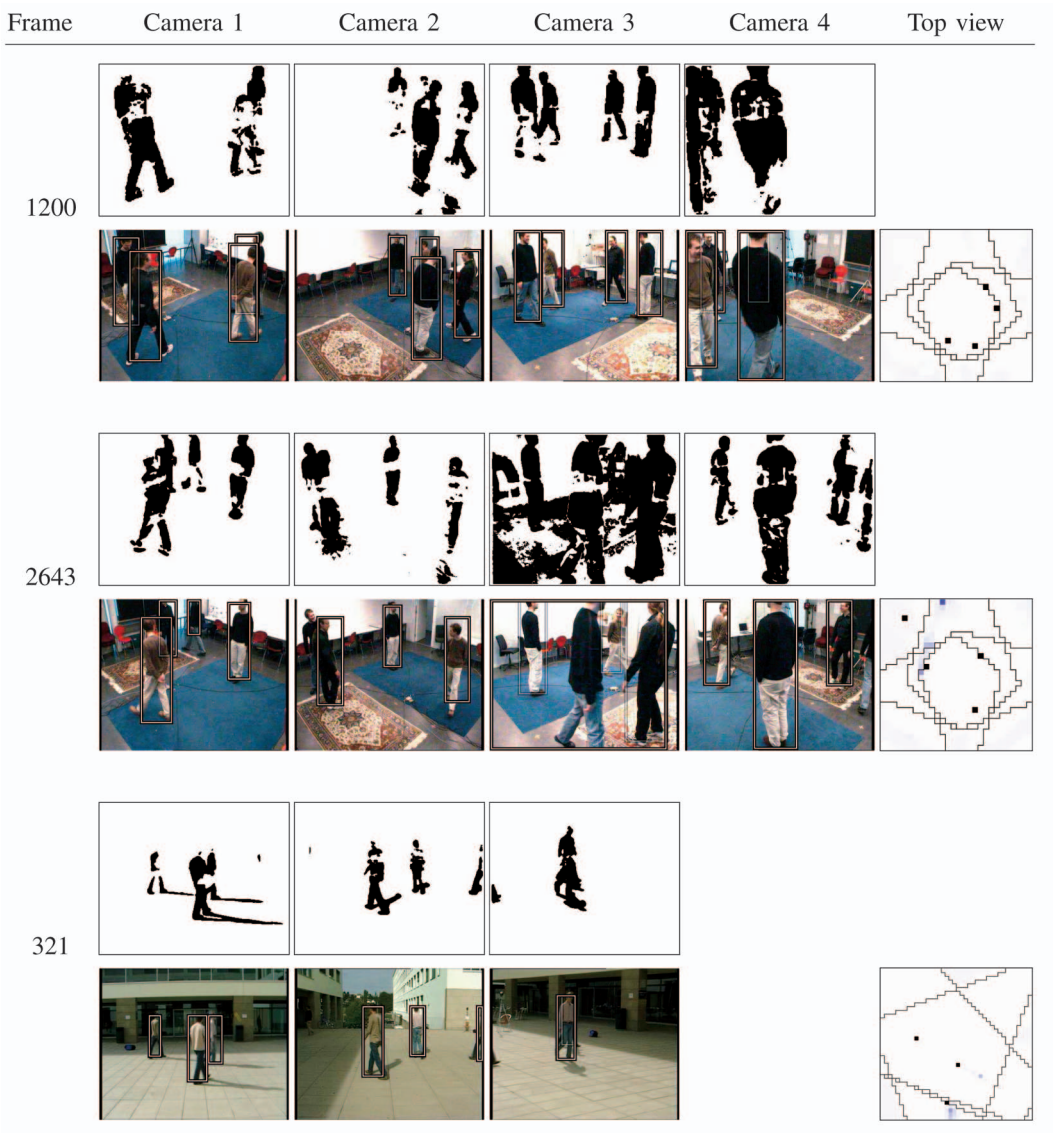

Fig. 8. Results of the POM estimation. Shown are the background subtraction images and the corresponding detection after thresholding. Displayed on the top right column are the POM, without any post processing. Time frame 2643 illustrates a failure case, in which the algorithm detects a person at the wrong place, due to the bad quality of background subtraction.

6.4.1 Indoor Sequences

On both of those sequences, the algorithm performs very well and does not lose any of the tracked persons. To investigate the spatial accuracy of our approach, we compared the estimated locations with the actual locations of the individuals present in the room as follows.

We picked 100 frames at random among the complete four individual sequence and marked by hand a reference point located on the belly of every person present in every camera view. For each frame and each individual, from that reference point and the calibration of the four cameras, we estimated a ground location. Since the 100 frames were taken from a sequence with four individuals entering the room successively, we obtained 354 locations.

We then estimated the distance between this ground truth and the locations estimated by the algorithm. The results are depicted by the bold curve in Fig. 2. More than 90 percent of those estimates are at a distance of less than 31 cm and 80 percent of less than 25 cm, which is satisfactory, given that the actual grid resolution is 20 cm in these series of experiments.

To test the robustness of our algorithm, for each camera individually, we randomly blanked out a given fraction of the images acquired by that camera. As a result, the frames, which are made of all the images acquired at the same time, could contain one or more blank images. This amounts to deliberately feeding the algorithm with erroneous information: Blank images provide incorrect evidence that there was no moving object in that frame and consequently degrades the accuracy of the occupancy estimate. Hence, this constitutes a stringent test of the effectiveness of optimizing the trajectories with dynamic programming. The accuracy remains unchanged for an erasing rate as high as 20 percent. The performance of the algorithm only starts to get noticeably worse when we get rid of 1/3 of the images, as shown in Fig. 2.

To investigate the robustness of the algorithm with respect to the number of cameras, we have performed tracking on the indoor sequence, with six people using only

Fig. 9. Results of the tracking algorithm. Each row displays several views of the same time frame coming from different cameras. The column for camera 4 is left blank for scenes for which only three cameras were used. Note that on frame 1689, people 2 and 3 are seen by only one camera but are nonetheless correctly identified.

two or three of the four available camera streams. When reducing to three cameras, there is no noticeable difference. When using only two cameras, the algorithm starts to incorrectly track people when more than four people enter the scene since two cameras only do not provide enough visual cues to resolve some of the occlusions.

6.4.2 Outdoor Sequences

Despite the disturbing influence of external elements such as shadows, a sliding door, cars passing by, tables and chairs in the middle of the scene, and the fact that people can enter and exit the tracked area from anywhere on some sequences, the algorithm performs well and follows people accurately. In many cases, because the cameras are not located ideally, individuals appear on one stream alone. They are still correctly localized due to both the time consistency and the rectangle matching of the ground occupancy estimation, which is able to exploit the size of the blobs, even in a monocular context. Such a case appears for individuals 2 and 3 in frame 1689, as shown in Fig. 9.

The algorithm is not disturbed by uncommon people behaviors such as people jumping or bending (see Fig. 10). It can also deal with the presence of static obstacles or strong shadows on the tracked area. As also illustrated by some outdoor sequences, a small number of people dressed the same color are correctly detected.

On very challenging sequences, which include at once more than five people, illumination changes, and similar color clothes, the algorithm starts to make mistakes and mixes some identities or fails to detect people. The main reason is that as the number of people increases, some people are both occluded on some camera views and out of the range of the other cameras. When this happens for too many consecutive time frames, the dynamic programming is not able to cope with it, and mistakes start to appear.

6.4.3 Rectangle Size Influence

We checked the influence of the size of the rectangular shapes that we use as models. The results of the POM algorithm are almost unchanged for model sizes between 1.7 m and 2.2 m. The performance tends to decrease for sizes noticeably smaller. This can be explained easily: if the model is shorter than the person, then the algorithm will be more sensitive to spurious binary blobs that it may explain by locating a person in the scene, which is less likely to happen with taller models.

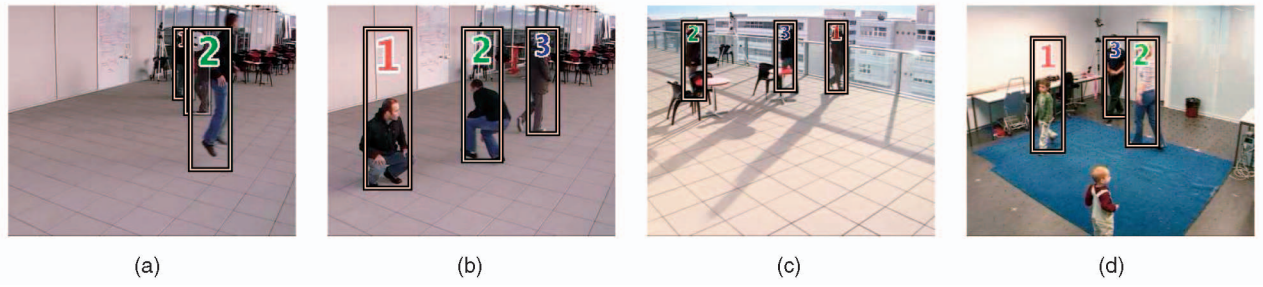

Fig. 10. Some of the difficulties that our algorithm is capable to deal with. (a) People jumping. (b) People bending. (c) Obstacles and very long and sharp shadows. (d) Small kids.

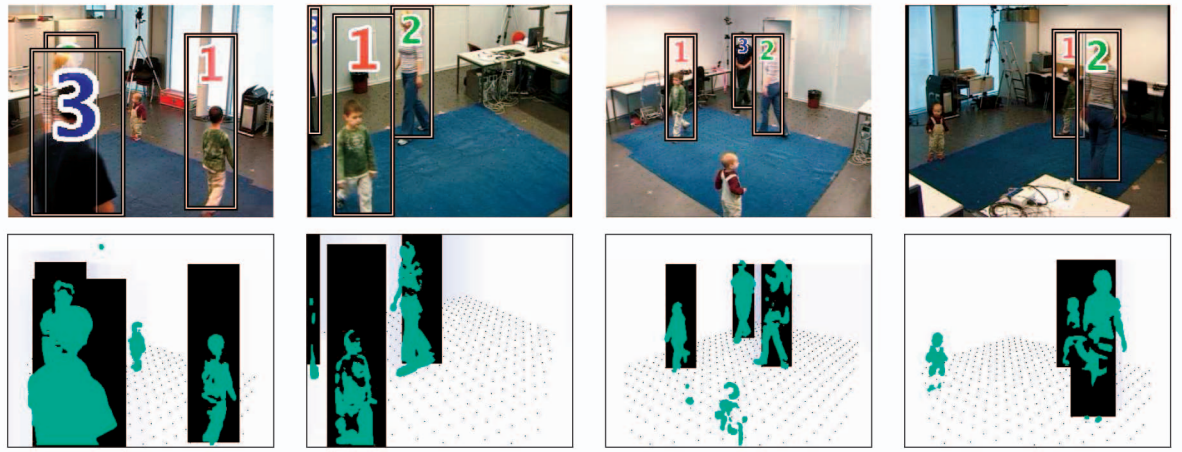

Fig. 11. Illustration of POM sensitivity to people size. First row shows the four camera views of a single time frame, as well as the detection results. Second row displays the synthetic images corresponding to our ground occupancy estimation, along with the background subtraction blobs. Although quite small, the boy is correctly tracked. The little girl, however, is too small to be detected.

Further investigation about our algorithm's sensitivity to people size has been carried out by using a video sequence involving two adults and two children. As illustrated in Fig. 11, throughout the whole sequence, the algorithm follows accurately the 4-year-old boy, who is about 1 m high. The 2-year-old girl, who measures less than 80 cm, is, however, not detected. The foreground blobs that she produces, about 1/4 in surface of an adult blob, are just too small for the POM algorithm to notice her.

6.5 Computational Cost

The computation involves two major tasks: estimating the POMs and optimizing the trajectories via dynamic programming.

Intuitively, given the iterative nature of the computation in Section 5, one would expect that the first task to be very expensive. However, the integral-image-based algorithm in Section 5.4 yields a very fast implementation. As a result, we can process a four-camera sequence at 6 fps on the average on a standard PC.

The second step relies on a very standard dynamic programming algorithm. Since the hidden state can take only $G \simeq 1{,}000$ values, and the motion model is highly peaked, the computation is fast, and we can, on the average, process 2 fps.

6.6 Limitations of the Algorithm

Although the full algorithm is very robust, we discuss here the limitations of its two components and the potential ways to overcome them.

6.6.1 *Computing the POM*

The quality of the occupancy map estimation can be affected by three sources of errors: The poor quality of the output of the background subtraction, the presence of people in an area covered by only one camera, and the excessive proximity of several individuals.

In practice, the first two difficulties only result in actual errors when they occur simultaneously, which is relatively rare and could be made even rarer by using a more sophisticated approach to background subtraction. The main weakness of the method that we currently use is that it may produce blobs that are too large due to reflections on walls or glossy floors. This does not affect the performance if an individual is seen in multiple views but may make him appear to be closer than he truly is, if seen in a single view. Similarly, shadows are segmented as moving parts and can either make actual silhouettes appear larger or create ghosts. However, the latter are often inconsistent with the motion model and are filtered out.

The third difficulty is more serious and represents the true limitation of our approach. When there are too many people in the scene for any background subtraction algorithm to resolve them as individual blobs in at least one view, the algorithm will fail to correctly detect and locate all the individuals. The largest number of people that we can currently handle is hard to quantify because it depends on the scene configuration, the number of cameras, and their exact locations. However, the results presented in

this section are close to the upper limit that our algorithm can tolerate with the specific camera configurations that we use. A potential solution to this problem would be to replace background subtraction by people detectors that could still respond in a crowd.

6.6.2 Computing the Optimal Trajectories

Due to the coarse discretization of the grid, we have to accept very fast motions between two successive frames to allow for realistic individual speed over several time frames. This could be overcome with a finer grid at greater computational cost. Also, we neither enforce motion consistency along the trajectories nor account for the interactions between people. Greater robustness to potential errors in the occupancy map estimates could therefore be achieved by representing richer state spaces for the people that we track and explicitly modeling their interactions. Of course, this would come at the cost of an increased computational burden.

7 Conclusion

We have presented an algorithm that can reliably track multiple persons in a complex environment and provide metrically accurate position estimates. This is achieved by combining the POM, which provides a very robust estimation of the occupancy on the ground plane in individual time frame, with a global optimization of the trajectories of the detected individuals over 100-frame batches. This optimization scheme introduces a 4 second delay between image acquisition and output of the results, which we believe to be compatible with many surveillance applications, given the robustness increase that it offers.

There are many possible extensions of this work. The most obvious ones are the improvements of our stochastic model. The color model could be refined by splitting bodies into several uniform parts instead of relying on an independence assumption, and we could calibrate the colors across cameras, as in [24], to combine more efficiently cues from different views. Similarly, the motion model could take into account consistency of speed and direction by increasing the state space with a coarse discretization of the motion vector. Modeling the avoidance strategies between people would also help. However, it is worth noting that even with our very simple models, we already obtain very good performance, thus underlining the power of our trajectory optimization approach.

Beside those straightforward improvements, a more ambitious extension would be to use the current scheme to automatically estimate trajectories from a large set of video sequences, from which one could then learn sophisticated appearance and behavior models. These models could, in turn, be incorporated into the system to handle the increasingly difficult situations that will inevitably arise when the scenes become more crowded, or we use fewer cameras.

Appendix A

Relation between the q_ls

We are looking for an approximation of $P(X_t^k = 1|\mathbf{B}_t)$. Having introduced a generative model of $P(\mathbf{B}_t|\mathbf{X}_t)$, we estimate the product law $Q(\mathbf{X}_t) = \prod_n Q(X_t^n)$ minimizing the Kullback-Leibler divergence to the true conditional law on $\mathbf{X}_t$, given $\mathbf{B}_t$, under this model. We denote by E_Q the expectation under $\mathbf{X} \sim Q$, and we derive the Kullback-Leibler divergence with respect to the unknown q_k:

$$\frac{\partial}{\partial q_k} KL(Q, P(\cdot|\mathbf{B})),$$

$$= \frac{\partial}{\partial q_k} E_Q\left(\log \frac{Q(\mathbf{X})}{P(\mathbf{X}|\mathbf{B})}\right), \tag{36}$$

$$= \frac{\partial}{\partial q_k} E_Q\left(\log \frac{Q(\mathbf{X})}{P(\mathbf{X})}\frac{P(\mathbf{B})}{P(\mathbf{B}|\mathbf{X})}\right), \tag{37}$$

$$= \frac{\partial}{\partial q_k} E_Q\left(\sum_l \log \frac{Q(X^l)}{P(X^l)} + \log P(\mathbf{B}) - \log P(\mathbf{B}|\mathbf{X})\right), \tag{38}$$

$$= \frac{\partial}{\partial q_k} E_Q\left(\log \frac{Q(X^k)}{P(X^k)} - \log P(\mathbf{B}|\mathbf{X})\right), \tag{39}$$

$$= \frac{\partial}{\partial q_k} q_k\left(\log \frac{q_k}{\epsilon_k} - E_Q\big(\log P(\mathbf{B}|\mathbf{X})|X^k = 1\big)\right),$$

$$+ \frac{\partial}{\partial q_k}(1-q_k)\left(\log \frac{1-q_k}{1-\epsilon_k} - E_Q\big(\log P(\mathbf{B}|\mathbf{X})|X^k = 0\big)\right), \tag{40}$$

$$= \log \frac{q_k}{\epsilon_k} + 1 - E_Q\big(\log P(\mathbf{B}|\mathbf{X})|X^k = 1\big)$$

$$- \log \frac{1-q_k}{1-\epsilon_k} - 1 + E_Q\big(\log P(\mathbf{B}|\mathbf{X})|X^k = 0\big), \tag{41}$$

$$= \log \frac{q_K(1-\epsilon_k)}{(1-q_k)\epsilon_k} - E_Q\big(\log P(\mathbf{B}|\mathbf{X})|X^k = 1\big)$$

$$+ E_Q\big(\log P(\mathbf{B}|\mathbf{X})|X^k = 0\big),$$

$$= \log \frac{q_K(1-\epsilon_k)}{(1-q_k)\epsilon_k} - E_Q\left(-\sum_c \Psi(B^c, A^c)\middle| X^k = 1\right)$$

$$+ E_Q\left(-\sum_c \Psi(B^c, A^c)\middle| X^k = 0\right). \tag{42}$$

Equality (36) is the definition of the Kullback-Leibler divergence, and (37) is obtained by applying Bayes' rule to $P(\mathbf{X}|\mathbf{B})$. Equality (38) is true under our assumption of independence of the X_ks, and (39) is true by removing terms that are constant with respect to q_k. We develop the expectation by conditioning on X_k to get (40), do formal differentiation to obtain (41), and finally introduce our model of $P(\mathbf{B}|\mathbf{X})$ and assumption of conditional independence of the B^c, given $\mathbf{X}$, to get (42).

Appendix B

Appearance Model

Recall that our appearance model is given by

$$P(\mathbf{I}_t|L_t^n = k), \tag{43}$$

where $\mathbf{I}_t$s are the input images at time frame t, and L_t^n is the random variable representing the location on the grid of individual n also at time t. From the input images $\mathbf{I}_t$, we use background subtraction to produce binary masks $\mathbf{B}_t$. We denote as $\mathbf{T}_t$ the colors of the pixels inside the blobs and treat the rest of the images as background, which is ignored.

Let X_k^t be a Boolean random variable standing for the presence of an individual at location k of the grid at time t. Then, we have

$$\overbrace{P(\mathbf{I}_t|L_t^n=k)}^{\text{Appearance model}} = \frac{P(\mathbf{I}_t)}{P(L_t^n=k)}P(L_t^n=k|\mathbf{I}_t), \quad (44)$$

$$\propto P(L_t^n=k|\mathbf{I}_t), \quad (45)$$

$$= P(L_t^n=k|\mathbf{B}_t,\mathbf{T}_t), \quad (46)$$

$$= P(L_t^n=k, X_t^k=1|\mathbf{B}_t,\mathbf{T}_t), \quad (47)$$

$$= P(L_t^n=k|X_t^k=1,\mathbf{B}_t,\mathbf{T}_t)P(X_t^k=1|\mathbf{B}_t,\mathbf{T}_t),$$

$$= \underbrace{P(L_t^n=k|X_t^k=1,\mathbf{T}_t)}_{\text{Color model}}\ \underbrace{P(X_t^k=1|\mathbf{B}_t)}_{\text{Ground plane occupancy}}. \quad (48)$$

Equality (44) follows directly from Bayes' formula. Equality (45) is true, since the probability of the image, without conditioning, does not depend on the trajectory, and the prior on the trajectories is flat. Equality (46) is true under the assumption that all information are carried by the product of the background subtraction and the set of the blob pixel colors. Equality (47) is true, since $L_t^n=k \Rightarrow X_t^k=1$ and, finally, (48) is true under the assumptions that the occupancy of a location X_t^k provides strictly more information about someone being at location k than the result of the background subtraction and that given the result of the background subtraction, the color of the blobs does not provide information about the occupancy.

APPENDIX C
COLOR MODEL

We have

$$\overbrace{P(L_t^n=k|X_t^k=1,\mathbf{T}_t)}^{\text{Color model}} = \frac{P(L_t^n=k, X_t^k=1,\mathbf{T}_t)}{P(X_t^k=1,\mathbf{T}_t)}, \quad (49)$$

$$= \frac{P(L_t^n=k, X_t^k=1,\mathbf{T}_t)}{\sum_m P(L_t^m=k, X_t^k=1,\mathbf{T}_t)}, \quad (50)$$

$$= \frac{P(L_t^n=k,\mathbf{T}_t)}{\sum_m P(L_t^m=k,\mathbf{T}_t)}, \quad (51)$$

$$= \frac{P(\mathbf{T}_t|L_t^n=k)}{\sum_m P(\mathbf{T}_t|L_t^m=k)}. \quad (52)$$

Equality (49) follows from Bayes' law, (50) is true by the complementarity of the events $L_t^m=k$, (51) is true since $L_t^m=k \Rightarrow X_k=1$, and finally, (52) is true by applying Bayes' law again.

ACKNOWLEDGMENTS

This work was supported in part by the Swiss Federal Office for Education and Science and in part by the Indo Swiss Joint Research Programme (ISJRP).

REFERENCES

[1] J. Black, T.J. Ellis, and P. Rosin, "Multi-View Image Surveillance and Tracking," *Proc. IEEE Workshop Motion and Video Computing,* 2002.

[2] D. Beymer, "Person Counting Using Stereo," *Proc. Workshop Human Motion,* pp. 127-133, 2000.

[3] J. Berclaz, F. Fleuret, and P. Fua, "Robust People Tracking with Global Trajectory Optimization," *Proc. Conf. Computer Vision and Pattern Recognition,* 2006.

[4] Q. Cai and J.K. Aggarwal, "Automatic Tracking of Human Motion in Indoor Scenes across Multiple Synchronized Video Streams," *Proc. Sixth IEEE Int'l Conf. Computer Vision,* 1998.

[5] R.T. Collins, "Mean-Shift Blob Tracking through Scale Space," *Proc. Conf. Computer Vision and Pattern Recognition,* p. 234, 2003.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean Shift," *Proc. Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 142-149, 2000.

[7] F. Fleuret, R. Lengagne, and P. Fua, "Fixed Point Probability Field for Complex Occlusion Handling," *Proc. 10th Int'l Conf. Computer Vision,* Oct. 2005.

[8] D. Focken and R. Stiefelhagen, "Towards Vision-Based 3D People Tracking in a Smart Room," *Proc. Fourth IEEE Int'l Conf. Multimodal Interfaces,* 2002.

[9] J. Giebel, D.M. Gavrila, and C. Schnorr, "A Bayesian Framework for Multi-Cue 3D Object Tracking," *Proc. Eighth European Conf. Computer Vision,* 2004.

[10] I. Haritaoglu, D. Harwood, and L. Davis, "Who, When, Where, What: A Real-Time System for Detecting and Tracking People," *Proc. Third IEEE Int'l Conf. Automatic Face and Gesture Recognition,* pp. 222-227, 1998.

[11] M. Han, W. Xu, H. Tao, and Y. Gong, "An Algorithm for Multiple Object Trajectory Tracking," *Proc. Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 864-871, June 2004.

[12] M. Isard and J. MacCormick, "Bramble: A Bayesian Multiple-Blob Tracker," *Proc. Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 34-41, July 2001.

[13] J. Kang, I. Cohen, and G. Medioni, "Tracking People in Crowded Scenes Across Multiple Cameras," *Proc. Sixth Asian Conf. Computer Vision,* 2004.

[14] J. Krumm, S. Harris, B. Myers, B. Brummit, M. Hale, and S. Shafer, "Multi-Camera Multi-Person Tracking for Easy Living," *Proc. Third IEEE Workshop Visual Surveillance,* 2000.

[15] S. Khan, O. Javed, and M. Shah, "Tracking in Uncalibrated Cameras with Overlapping Field of View," *Proc. Second IEEE Workshop Performance Evaluation of Tracking and Surveillance,* 2001.

[16] S. Khan and M. Shah, "Tracking People in Presence of Occlusion," *Proc. Fourth Asian Conf. Computer Vision,* 2000.

[17] V.I. Morariu and O.I. Camps, "Modeling Correspondences for Multi-Camera Tracking Using Nonlinear Manifold Learning and Target Dynamics," *Proc. Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 545-552, 2006.

[18] A. Mittal and L. Davis, "M2tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," *Int'l J. Computer Vision,* vol. 51, no. 3, pp. 189-203, 2003.

[19] I. Mikic, S. Santini, and R. Jain, "Video Processing and Integration from Multiple Cameras," *Proc. DARPA Image Understanding Workshop,* 1998.

[20] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Crowded Environment," *Proc. Conf. Computer Vision and Pattern Recognition,* 2004.

[21] K. Otsuka and N. Mukawa, "Multi-View Occlusion Analysis for Tracking Densely Populated Objects Based on 2D Visual Angles," *Proc. Conf. Computer Vision and Pattern Recognition,* 2004.

[22] N.M. Oliver, B. Rosario, and A.P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 831-843, Aug. 2000.

[23] K. Okuma, A. Taleghani, N. de Freitas, J.J. Little, and D.G. Lowe, "A Boosted Particle Filter: Multitarget Detection and Tracking," *Proc. Eighth European Conf. Computer Vision,* May 2004.

[24] F. Porikli and A. Divakaran, "Multi-Camera Calibration, Object Tracking and Query Generation," *Proc. Int'l Conf. Multimedia and Expo,* vol. 1, pp. 635-6569, 2003.

[25] K. Smith, D. Gatica-Perez, and J.-M. Odobez, "Using Particles to Track Varying Numbers of Interacting People," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2005.

[26] D.B. Yang, H.H. Gonzáles-Baños, and L.J. Guibas, "Counting People in Crowds with a Real-Time Network of Simple Image Sensors," *Proc. Ninth Int'l Conf. Computer Vision,* pp. 122-129, 2003.
[27] F. Ziliani and A. Cavallaro, "Image Analysis for Video Surveillance Based on Spatial Regularization of a Statistical Model-Based Change Detection," *Proc. 10th Int'l Conf. Image Analysis and Processing,* 1999.

François Fleuret received the PhD degree in probability from the University of Paris VI in 2000. After one year at the Department of Computer Science, University of Chicago, he was hired as a full researcher at the French National Institute for Research in Computer Science and Control (INRIA). In 2004, he joined the Computer Vision Laboratory, Swiss Federal Institute of Technology (CVLab EPFL), where he spent three years, before joining the Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP) in 2007 as a senior researcher in machine learning. His main research interests are at the interface between statistical methods and algorithmic, centered on the development of algorithmically efficient machine learning techniques.

Jérôme Berclaz received the MS degree in communication systems from the Swiss Federal Institute of Technology (EPFL) in 2004. He is now pursuing the PhD degree at the Computer Vision Laboratory, Swiss Federal Institute of Technology (CVLab EPFL). His main research interest is computer vision, with emphasis on tracking.

Richard Lengagne received the PhD degree in computer science from the University of Paris XI, France, in 2000. Prior to that, he successively worked at ATR International, Kyoto, Japan, the Chinese Academy of Sciences, Beijing, the Stanford Research Institute, Menlo Park, USA, and the French National Institute for Research in Computer Science and Control (INRIA), Rocquencourt, France. He participated in various research activities ranging from speech recognition to medical image processing and three-dimensional object reconstruction from stereo. He then joined the Computer Graphics Laboratory and the Computer Vision Laboratory, Swiss Federal Institute of Technology (EPFL), Switzerland, where he took part in human body tracking projects. He is currently with the R&D Department of GE-Security, VisioWave, where he is involved in the development of visual surveillance products. He authored or coauthored around 15 papers published in international computer vision conferences and journals.

Pascal Fua received a degree from Ecole Polytechnique, Paris, in 1984 and the PhD degree in computer science from the University of Orsay in 1989. He joined the Swiss Federal Institute of Technology (EPFL) in 1996, where he is now a professor in the School of Computer and Communication Science. Before that, he worked at SRI International and at the French National Institute for Research in Computer Science and Control (INRIA), Sophia-Antipolis, as a computer scientist. He is a member of the editorial board of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and has been a program committee member and area chair of several major vision conferences. His research interests include human body modeling from images, optimization-based techniques for image analysis and synthesis, and using information theory in the area of model-based vision. He has (co)authored more than 100 publications in refereed journals and conferences. He is a senior member of the IEEE.