Robust People Tracking with Global Trajectory Optimization*

Jérôme Berclaz François Fleuret Pascal Fua
EPFL – CVLAB
CH – 1015 Lausanne, Switzerland
{jerome.berclaz,francois.fleuret,pascal.fua}@epfl.ch

Abstract

Given three or four synchronized videos taken at eye level and from different angles, we show that we can effectively use dynamic programming to accurately follow up to six individuals across thousands of frames in spite of significant occlusions. In addition, we also derive metrically accurate trajectories for each one of them.

Our main contribution is to show that multi-person 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 avoid confusing them with one another. In this way, we achieve robustness by finding optimal trajectories over many frames while avoiding the combinatorial explosion that would result from simultaneously dealing with all the individuals.

1. Introduction

In this paper, we show that we can effectively use dynamic programming in situations such as those depicted by Fig. 1 to keep track of people who occlude each other. This results in a fully automated system that can track up to 6 people in a room for several minutes using only four cameras, without producing any false positives or false negatives in spite of severe occlusions and lighting variations. As shown in Fig. 1, our system also provides location estimates that are accurate to within a few tens of centimeters.

We combine probabilities of occupancy of the ground plane that are computed at each time step independently [3] with color and motion models that let us enforce temporal continuity. 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 global optima of scores summed over many frames. This gives us great robustness to data loss: As shown in Fig. 1, there is no measurable performance decrease if as many as 20% of the images are lost, and only a small one if 30% are.

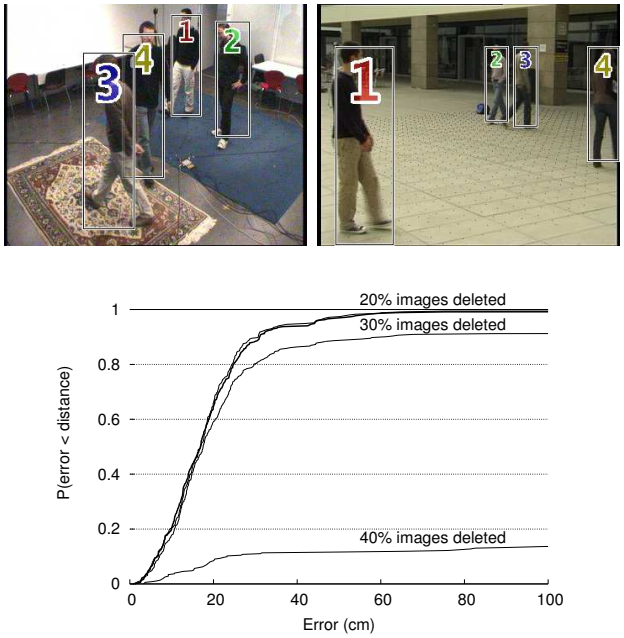

Figure 1. Top row: tracking results. Bottom row: cumulative distributions of the position estimate error on a 3800-frame sequence. See §6.1 for details.

More specifically, we process the video sequences by batches of one hundred frames and use dynamic programming to compute the most likely trajectory of each individual. This batch processing introduces a 4s delay, but this is quite acceptable for many surveillance applications. To achieve consistency over the whole sequence, we only keep the result on the ten first frames and slide our temporal window by ten frames. To handle entrances and departures, we consider a virtual hidden location containing a very large number of people, each with a very small probability of entering the visible scene. Our mathematical framework treats the visible and hidden individuals similarly and en-

*This work was supported in part by the Swiss Federal Office for Education and Science and by the Swiss National Science Foundation

trances occur when image data makes the optimal trajectory of someone located in the hidden location cross into the visible space.

Our main contribution is to show that multi-person tracking can be reliably achieved by processing individual trajectories separately over long sequences, given that a reasonable heuristic is used to rank these individuals and avoid confusing them with one another. Processing trajectories individually lets us 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. This is what lets us compute trajectories that are optimal over many frames.

2. Related Work

State-of-the-art methods can be divided into monocular and multi-view approaches that we briefly review in this section. While our own method shares many features with these techniques, it differs in two important respects. First, we rely on dynamic programming to ensure greater stability in challenging situations by simultaneously taking into account multiple frames. Second, it relies on a discretization of the full area of interest, and is therefore able to deal with very flat distributions.Finally, our approach combines the usual color and motion models with a sophisticated estimation of the probability of occupancy.

2.1. Monocular approaches

Approaches that perform tracking in a single view prior to computing correspondences across views typically rely on extracting groups of pixels, which can then be assigned to individual people [6, 2, 8]. Tracking performance can be significantly increased by taking color into account. For example, in [9], the images are segmented pixel-wise 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 under occlusion. When such a case occurs, models of visible persons keep being updated, but the update of occluded ones stops. This may cause trouble if their appearances have changed noticeably when they reemerge.

More recently, multiple humans have been simultaneously detected and tracked in crowded scenes [16] using Monte-Carlo-based methods to estimate their number and positions. In [13], multiple people are also detected and tracked in front of complex backgrounds using mixture particle filters guided by people models learnt by boosting. In [5], multi-cue 3D object tracking is addressed by combining particle-filter based Bayesian tracking and detection using learnt spatio-temporal shapes. This approach leads to impressive results but requires shape, texture, and stereo information as input. Finally [15] proposes a particle-filtering scheme with a MCMC optimization which handles naturally entrances and departures, and introduces a finer modeling of interactions between individuals as a product of pairwise potentials.

2.2. Multi-view 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 may be facilitated by the use of 2 sets of stereo color cameras[10]. However, in most approaches that only take a set of 2D views as input, occlusion is mainly handled using the temporal consistency brought by a motion model, whether from Kalman filtering or more general Markov models. As a result, these approaches may not always be able to recover if the process starts diverging.

Blob-based Methods In [11], 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. 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 [4], a best-hypothesis and a multiple-hypothesis approaches are compared to find people tracks from 3D locations obtained from foreground binary blobs extracted from multiple calibrated views. In [14], silhouette-based visual angles are obtained from motion blobs. In case of occlusion ambiguities, multiple occlusion hypotheses are generated given predicted object states and previous hypotheses. A Bayesian framework is applied to test multiple hypotheses using a state transition model, a dynamics model for transitions between occlusion structures and the measurements.

Color-Based Methods [12] proposes a system that segments, detects and tracks multiple people in a scene 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 using a Kalman filter.

In [7], 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.

Table 1. Notations

We use bold letters for vectors and drop the indices to denote a vector of values corresponding to several values of the said indices, for example $\mathbf{L}_t$ and $\mathbf{L}^n$ below.

C	number of cameras
G	number of locations in the ground discretization ($\simeq 1000$)
T	number of frames processed in one batch ($T = 100$)
t	frame index
$\mathbf{I}_t$	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
N^*	virtual number of people, including the non-visible ones
$\mathbf{L}_t$	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, $\mathbf{L}^n = (L_1^n, \ldots, L_T^n)$
μ_n^c	color distribution of individual n from camera c
X_t^k	boolean random variable standing for the occupancy of location k on the ground plane $\left(X_t^k = 1\right) \Leftrightarrow (\exists q, L_t^q = k)$

3. Overview and Notations

Here, we give a short overview of the complete algorithm, before going into more details in the following section. From now on, we will use the notations summarized by Table. 1.

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

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. Assuming that the visible part of the ground plane has been discretized into a finite number G of regularly spaced 2–D locations, the L_t^n variables take discrete values in the range $\{1, \ldots, G,\ \mathcal{H}\}$, where $\mathcal{H}$ denotes a hidden location. 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 individual does not change the probability of a new individual entering the scene.

Given $\mathbf{I}_t$, our task is therefore to find the values of the $\mathbf{L}_t$ that maximize $P(\mathbf{L}_1, \ldots, \mathbf{L}_T \,|\, \mathbf{I}_1, \ldots, \mathbf{I}_T)$.

3.1. Stochastic Modeling

Our optimization scheme optimizes trajectories successively, and the optimization of an individual trajectory relies on an appearance model and a motion model.

The appearance model $P(\mathbf{I}_t \,|\, L_t^n = k)$ is a combination of two terms. The first is an estimate of the probability of occupancy of the ground plane that is computed at each time step independently [3] given the output of a simple background subtraction algorithm. It is depicted by Fig. 2. The second is a very generic color-histogram based model for each individual. Note that the ground plane occupancy estimate says 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.

The motion model $P(L_{t+1}^n \,|\, L_t^n = l)$ is simply a distribution into a disc of limited radius, 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.

3.2. Optimization

Given this model, we compute the optimal trajectories over the whole batch, 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 if our algorithm connected 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 other ones. This order also ensures that if an individual remains in the hidden location, all the other people present in the hidden location will also stay there, and therefore do not need to be processed.

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.

4. Stochastic Modeling

We compute the MAP of $P(\mathbf{L}_1, \ldots, \mathbf{L}_T \,|\, \mathbf{I}_1, \ldots, \mathbf{I}_T)$ by processing trajectories individually. We show in §5.1 that this requires only modeling at a given frame t the conditional distribution $P(\mathbf{I}_t \,|\, L_t^n = k)$ of the images given the location of one individual. We describe this modeling in the present section.

From the input images $\mathbf{I}_t$, we use background subtraction to produce binary masks $\mathbf{B}_t$ and the pixels inside the blobs

$\mathbf{T}_t$. The rest of the images is treated as background and ignored. We have:

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

where $P(L_t^n = k \,|\, X_t^k = 1,\, \mathbf{T}_t)$ is based on the color model and $P(X_t^k = 1 \,|\, \mathbf{B}_t)$ is an estimate of the ground plane occupancy.

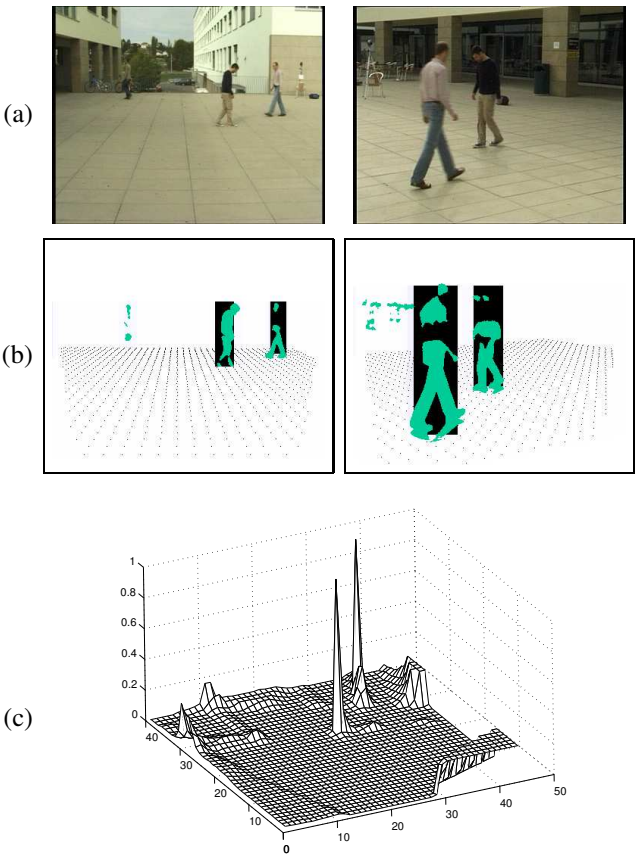

Figure 2. Original images from two cameras (a), binary images produced by background subtraction (green) and synthetic average images computed from them (b). The surface on (c) represents the corresponding occupancy probabilities ρ_k on the grid.

4.1. Estimating Ground Plane Occupancy

The first module of our tracking algorithm is a frame-by-frame people detector that takes as input the binary masks $\mathbf{B}_t$ generated by a simple background subtraction algorithm and computes for each location in the ground plane the conditional marginal probability of presence of an individual. To this end, we have slightly improved our earlier Fixed-Point Probability Field (FPPF) algorithm [3] by including ad hoc descriptions of potentially moving parts of the background, such as sliding doors. This is legitimate since the camera environment is fixed and known.

After discretization of the ground plane into a regular grid (Fig. 2.b), this algorithm provides for every location k with an estimate of

$$\rho_k = P(X_t^k = 1 \,|\, B_t^1, \ldots, B_t^C) \tag{1}$$

where X_t^k stands for the occupancy of location k at time t by any individual.

The correspondence between ground-plane locations and the camera views is provided by the mean, for every camera, of a collection of rectangles standing for human shapes located at every position of the grid (Fig. 2.b). Those rectangles are computed from the average human height and the homography mapping the ground plane in the camera view.

4.2. Color model

We assume that if somebody is present at a certain location k, her presence influences the color of the pixels located at the intersection of the moving blobs and the rectangle corresponding to the location k. We model that dependency as if the pixels were independent and identically distributed and followed a density in the RGB space associated to the individual.

If an individual was present in the frames preceding the current batch, we have an estimation of her distribution, since we have previously collected the pixels in all frames at the locations of her estimated trajectory. If she is at the hidden location $\mathcal{H}$, her color distribution is flat.

Let $T_t^c(k)$ 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 corresponding to location k in that same field of view (Fig. 2.b).

Let $\mu_1^c, \ldots, \mu_{N^*}^c$ be the color distributions of the N^* individuals present in the scene at the beginning of the current batch of T frames, for camera c. We have

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

where

$$\begin{aligned} P(\mathbf{T}_t \,|\, L_t^n = k) &= P(T_t^1(k), \ldots, T_t^C(k) \,|\, L_t^n = k) \\ &= \prod_{c=1}^{C} \prod_{\rho \in T_t^c(k)} \mu_n^c(\rho) \end{aligned}$$

4.3. Motion model

We opted for a very unconstrained and simple motion model $P(L_t = k \,|\, L_{t-1} = \tau)$. It simply limits the maximum speed allowed for the tracked people by being zero for $||k - \tau||$ greater than a maximum distance and constant otherwise. We chose a tolerant maximum distance of one square of the grid per frame, which corresponds to a speed of almost 12mph. We also defined explicitly 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.

5. Optimization

We first describe how we compute the optimal trajectory of a person given a batch of images. We then describe the whole optimization scheme that processes trajectories one after another and heuristically chooses an adequate processing order.

5.1. Single trajectory

We consider in the following only the trajectory $\mathbf{L}^n = (L_1^n, \ldots, L_T^n)$ of individual n over T frames. We are looking for the trajectory $(l_1^n, \ldots, l_T^n)$, taking values in $\{1, \ldots, G, \mathcal{H}\}$ where $\mathcal{H}$ is a hidden location. 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 she is not. The score to maximize is

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

If we introduce the maximum of the probability of both the observations and the most probable trajectory ending up at location k at time t

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

we can use the well-known Viterbi algorithm

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

to perform a global search with dynamic programming.

5.2. Multiple trajectories

Given a batch of T frames $\mathbf{I} = (\mathbf{I}_1, \ldots, \mathbf{I}_T)$, we want to maximize the posterior conditional probability

$$P(\mathbf{L}^1 = \mathbf{l}^1, \ldots, \mathbf{L}^{N^*} = \mathbf{l}^{N^*} \,|\, \mathbf{I}).$$

We assume that optimizing trajectories altogether is the same as optimizing one trajectory after another, provided that it is done in an adequate order. We are thus looking for

$$\begin{aligned} \hat{\mathbf{l}}^1 &= \arg\max_l P(\mathbf{L}^1 = l \,|\, \mathbf{I}), \\ \hat{\mathbf{l}}^2 &= \arg\max_l P(\mathbf{L}^2 = l \,|\, \mathbf{I}, \mathbf{L}^1 = \hat{\mathbf{l}}^1), \\ &\vdots \\ \hat{\mathbf{l}}^{N^*} &= \arg\max_l P(\mathbf{L}^{N^*} = l \,|\, \mathbf{I}, \mathbf{L}^1 = \hat{\mathbf{l}}^1, \mathbf{L}^2 = \hat{\mathbf{l}}^2, \ldots). \end{aligned}$$

Such a procedure is correct under the assumption that a term of the form $P(\mathbf{L}^n = l \,|\, \mathbf{L}^1 = \hat{\mathbf{l}}^1, \ldots, \mathbf{L}^{n-1} = \hat{\mathbf{l}}^{n-1}, \mathbf{I})$ can not be substantially increased by choosing different trajectories $\hat{\mathbf{l}}^1, \ldots, \hat{\mathbf{l}}^{n-1}$, at least not enough to change the maximum. This is true in our case, as long as the trajectories $\hat{\mathbf{l}}^1, \ldots, \hat{\mathbf{l}}^{n-1}$ do not steal locations useful to $\hat{\mathbf{l}}^n$. We ensure that property by using an heuristic to rank the processing of the individuals. Note that under our model we have

$$\begin{aligned} &P(\mathbf{L}^n = l \,|\, \mathbf{I}, \mathbf{L}^1 = \hat{\mathbf{l}}^1, \ldots, \mathbf{L}^{n-1} = \hat{\mathbf{l}}^{n-1}) \\ &\qquad = P(\mathbf{L}^n = l \,|\, \mathbf{I}, \forall k < n, \forall t, L_t^n \neq \hat{l}_t^k), \end{aligned}$$

which can be seen as $P(\mathbf{L}^n = l \,|\, \mathbf{I})$ with a reduction of the admissible locations in the grid.

We first extend the trajectories that have been found with confidence in the previous batches. We then process the lower confidence ones. As a result, a low probability trajectory, that 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 spatial constraints on such a problematic trajectory when we finally get around to modeling it.

To this end, we use as a ranking score the concordance of the estimated trajectories in the previous batches and the localization cue provided by FPPF. Since there is a high degree of overlapping between successive batches, the challenging segment of a trajectory – due to failure of the background subtraction or change in illumination for instance – is met in several batches before it actually happens during the ten kept frames. Thus, the heuristic would have ranked the corresponding individual in the last ones to be processed when the problem occurs.

This heuristic naturally pushes the trajectories starting in the hidden location $\mathcal{H}$ – those not visible in the first frame of the batch – to the end of the computation. The algorithm does not actually compute all the N^* trajectories: It stops

as soon as one of the processed one remains in the hidden location for the complete batch of frames, since all other not-yet-processed individuals are identical and would do the same.

6. Results

We estimated the performance of our algorithm on several sequences shot indoor with four cameras and outdoor with three cameras. The indoor sequences involve up to six people and trajectories more complex than what happens usually in real-life situations. The outdoor sequences were shot on our campus and involve people going about their normal business, whose trajectories are actually simpler. In all our experiments, the cameras are mounted at, or just above, head level, and many occlusions occur.

Because the observed area is discretized into a finite number of positions, we linearly interpolate the trajectories on the output images to smooth them.

6.1. Indoor sequences

The indoor sequences were shot by a video-surveillance dedicated setup of 4 synchronized cameras in a 50m^2 room. Two cameras were roughly at head level ($\simeq 1.80$m) and the two others slightly higher ($\simeq 2.30$m). They were located at each corner of the room. The sequences are about 3000 frames long and involve up to six individuals.

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

On all those sequences, the algorithm performs very well and does not lose a single one 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 computed the distance between this ground-truth and the locations estimated by the algorithm. The results are depicted by the bold curve on Fig. 1. More than 90% of those estimates are at a distance of less than 31cm and 80% of less than 25cm. We also computed similar curves after having replaced a certain percentage of images taken randomly over the complete sequence by blank images. The accuracy remains unchanged for an erasing rate as high as 20%. The performance of the algorithm starts to get worse when we get ride of one third of the images, as shown with the thin curves on Fig. 1.

6.2. Outdoor sequences

The outdoor sequences were shot in front of the entrance of a building on our campus. We used three 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 three angles. The ground is flat with a regular pavement.

The area of interest is of size $10\text{m} \times 10\text{m}$ and discretized into $G = 40 \times 40 = 1600$ locations, corresponding to a regular grid with a resolution of 25cm. Up to four individuals appear simultaneously. Despite disturbing influence of external elements such as shadows, a sliding door, cars passing by, and the fact that people can enter and exit the tracked area from anywhere, 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 FPPF, which is able to exploit the size of the blobs even in a monocular context. On outdoor sequences as well, the algorithm does not produce one false positive or false negative, nor make confusion between individuals.

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 through global optimization of their trajectories over 100-frame batches. This 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 it offers.

There are many possible extensions of this work. The most obvious ones are improvements of our stochastic model. The color model could be refined by splitting bodies into several uniform parts instead of relying on the i.i.d. assumption. Similarly, the motion model could take into account consistency of speed and direction. Modeling the avoidance strategies between people would also help.

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, from which one could then learn sophisticated behavior models.

References

[1] J. Black, T. Ellis, and P. Rosin. Multi-view image surveillance and tracking. In *IEEE Workshop on Motion and Video Computing*, 2002.

[2] Q. Cai and J. Aggarwal. Automatic tracking of human motion in indoor scenes across multiple synchronized video streams. In *International Conference on Computer Vision*, 1998.

Figure 3. Results of the tracking algorithm. Each row displays several views of the same time frame coming from different cameras. The last image in a row is the top view of the observed area, with the corresponding locations of the people tracked.

[3] F. Fleuret, R. Lengagne, and P. Fua. Fixed point probability field for complex occlusion handling. In *International Conference on Computer Vision*, Beijing, China, October 2005.

[4] D. Focken and R. Stiefelhagen. Towards vision-based 3d people tracking in a smart room. In *IEEE International Conference on Multimodal Interfaces*, 2002.

[5] J. Giebel, D. Gavrila, and C. Schnorr. A bayesian framework for multi-cue 3d object tracking. In *Proceedings of European Conference on Computer Vision*, 2004.

[6] I. Haritaoglu, D. Harwood, and L. Davis. Who, when, where, what: A real time system for detecting and tracking people. In *Automated Face and Gesture Recognition*, pages 222–227, 1998.

[7] J. Kang, I. Cohen, and G. Medioni. Tracking people in crowded scenes across multiple cameras. In *Asian Conference on Computer Vision*, 2004.

[8] S. Khan, O. Javed, and M. Shah. Tracking in uncalibrated cameras with overlapping field of view. In *2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.

[9] S. Khan and M. Shah. Tracking people in presence of occlusion. In *Asian Conference on Computer Vision*, 2000.

[10] J. Krumm, S. Harris, B. Myers, B. Brummit, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easy living. In *Third IEEE Workshop on Visual Surveillance*, 2000.

[11] I. Mikic, S. Santini, and R. Jain. Video processing and integration from multiple cameras. In *Proceedings of the 1998 Image Understanding Workshop, Morgan-Kaufman, San Francisco*, 1998.

[12] A. Mittal and L. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203, 2003.

[13] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: multitarget detection and tracking. In *European Conference on Computer Vision*, Prague, Czech Republic, May 2004.

[14] K. Otsuka and N. Mukawa. Multi-view occlusion analysis for tracking densely populated objects based on 2-d visual angles. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, 2004.

[15] K. Smith, D. Gatica-Perez, and J.-M. Odobez. Using particles to track varying numbers of interacting people. In *Conference on Computer Vision and Pattern Recognition*, 2005.

[16] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, 2004.