Multi-Camera Tracking and Atypical Motion Detection with Behavioral Maps

Jérôme Berclaz[1⋆], François Fleuret[2⋆⋆], and Pascal Fua[1]

[1] Computer Vision Laboratory, EPFL, Lausanne, Switzerland
[2] IDIAP Research Institute, Martigny, Switzerland
jerome.berclaz@epfl.ch, francois.fleuret@idiap.ch, pascal.fua@epfl.ch

Abstract. We introduce a novel behavioral model to describe pedestrians motions, which is able to capture sophisticated motion patterns resulting from the mixture of different categories of random trajectories. Due to its simplicity, this model can be learned from video sequences in a totally unsupervised manner through an Expectation-Maximization procedure.
When integrated into a complete multi-camera tracking system, it improves the tracking performance in ambiguous situations, compared to a standard *ad-hoc* isotropic Markovian motion model. Moreover, it can be used to compute a score which characterizes atypical individual motions. Experiments on outdoor video sequences demonstrate both the improvement of tracking performance when compared to a state-of-the-art tracking system and the reliability of the atypical motion detection.

1 Introduction

Tracking multiple people in crowded scenes can be achieved without explicitly modeling human behaviors [1–5] but can easily fail when their appearances become too similar to distinguish one person from another, for example when two individuals are dressed similarly. Kalman filters and simple Markovian models have been routinely used for this purpose but do not go beyond capturing the continuous and smooth aspect of people's trajectories [6]. This problem has long been recognized in the Artificial Intelligence community and the use of much more sophisticated Markovian models, such as those that involve a range of strategies that people may pursue [7], have been demonstrated. For such an approach to become practical, however, the models have to be learnable from training data in an automated fashion instead of being painstakingly hand-crafted.

In this paper, we introduce models that can both describe how people move on a location of interest's ground plane, such as a cafeteria, a corridor, or a

⋆ supported in part by the Indo Swiss Joint Research Programme (ISJRP), and in part by funds of the European Commission under the IST-project 034307 DYVINE.
⋆⋆ supported by the Swiss National Science Foundation under the National Centre of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2).

train station, and be learned from image data. To validate these models, we use a publicly available implementation [8] of a multi-camera multi-people tracking system [2] first to learn them and second to demonstrate that they can help disambiguate difficult situations. We also show that, far from forcing everyone to follow a scripted behavior, the resulting models can be used to detect abnormal behaviors, which are defined as those that do not conform to our expectations. This a crucial step in many surveillance applications whose main task is to raise an alarm when people are having dangerous or prohibited behavior.

We represent specific behaviors by a set of *behavioral maps* that encode, for each ground plane location, the probability of moving in a particular direction. We then associate to people being tracked a probability of acting according to an individual map and to switch from one to the other based on their location. The maps and model parameters are learned by Expectation-Maximization in a completely unsupervised fashion. At run-time, they are used for robust and near real-time recovery of trajectories in ambiguous situations. Also, the same maps are used for efficient detection of abnormal behavior by computing the probability of retrieved trajectories under the estimated model.

The contribution of this paper is therefore to show that the models we propose are both sophisticated enough to capture higher-level behaviors that basic Markovian models cannot, and simple enough to be learned automatically from training data.

2 Related Works

With the advent of video surveillance and real-time people tracking algorithms, we have recently seen an increasing amount of research focused on acquiring spatio-temporal patterns by passive observation of video sequences[9–13].

Our approach shares similarities with [9], since we try to learn trajectory distributions from data as they do. However, while they model the trajectories in the camera view, and handle the temporal consistency using an artificial neural network with a short memory, we propose a more straight-forward modeling under a classical Markovian assumption with an additional behavioral hidden state. The metric homogeneity of the top-view allows for simpler priors, and the resulting algorithm can be integrated seamlessly in a standard HMM-based tracking system.

In a relatively close spirit, [10] uses an adaptive background subtraction algorithm to collect patterns of motion in the camera view. With the help of vector quantization, they build a codebook of representations out of this data, which they use to detect unusual events. [12] proceeds in a similar fashion to gather statistics from an online surveillance system. Using this data, they infer higher level semantics, such as the locations of entrance points, stopping areas, etc.

More related to our approach is the work of [11], which applies an EM algorithm to cluster trajectories recorded with laser-range finders. From this data, they derive an HMM to predict future position of the people. The use of laser-range scanners and their trajectory cluster model makes this approach more

adapted to an indoor environment where people have a relatively low freedom of movement, whereas our proposed behavioral maps are more generic and learned from standard video sequences shot with off the shelf cameras.

A quite different strategy has been chosen by [14]. In their work, they propose a generic behavior model for pedestrian based on discrete choice models, and apply it for reinforcing tracking algorithms. As opposed to our method, their framework does not need any training phase, but it is not able to learn the intrinsic specificity of a particular location.

Finally, our approach to handling human behaviors can be seen as a simplified version of Artificial Intelligence techniques, such as Plan Recognition [7] where the strategies followed by the agents are encoded by the behavioral maps. This simplification is what lets us learn our models from real data without having to hand-design them, which is a major step-forward with respect to traditional Artificial Intelligence problems.

3 Algorithm

We present in this section the core algorithm of our approach, first by describing the formal underlying motion model, and second by explaining both the E-M training procedure and the method through which the adequate training data was collected.

3.1 Motion Model

As described briefly in § 1, our motion model relies on the notion of behavioral map, a finite hidden state associated to every individual present in the scene. The rational behind that modeling is that an individual trajectory can be described with a deterministic large scale trajectory both in space and time (i.e. "he is going from door A to door B", "he is walking towards the coffee machine") combined with additional noise. The noise itself, while limited in scale, is highly structured: motion can be very deterministic in a part of a building where people do not collide, and become more random in crowded area. Hence this randomness is both strongly anisotropic – people in a certain map go in a certain direction – and strongly non-stationary – depending on their location in the area of interest the fluctuations differ. With an adequate class of models for individual maps, combining several of them allows for encoding such a structure.

Hence, re-using the formalism of [2], we associate to each individual a random process (L_t, M_t) indexed by the time t and taking its values in $\{1, \dots, G\} \times \{1, \dots, M\}$ where $G \simeq 1000$ is the number of locations in the finite discretization of the area of interest and M is the total number of behavioral maps we consider, typically less than 5. We completely define this process by first making a standard Markovian assumption, and then choosing models for both $P(L_0, M_0)$ and

$$P(L_{t+1}, M_{t+1} \,|\, L_t, M_t) \ . \tag{1}$$

Note that the very idea of maps strongly changes the practical effect of the Markovian assumption. For instance, by combining two maps that encode motions in opposite directions and a very small probability of switching from one map to the other, the resulting motion model is a mixture of two flows of individuals, each strongly deterministic. By making the probabilities of transition depend on the location, we can encode behaviors such as people changing their destination and doing a U-turn only at certain locations. Such a property can be very useful to avoid confusion of the trajectories of two individuals walking in opposite directions.

To define precisely (1), we first make an assumption of conditional independence between the map and the location at time $t+1$ given the same at time t $P(L_{t+1}, M_{t+1} \,|\, L_t, M_t) = P(L_{t+1} \,|\, L_t, M_t) P(M_{t+1} \,|\, L_t, M_t)$.

Due to the 20cm spatial resolution of our discretization, we have to consider a rather coarse time discretization to be able to model motion accurately. If we were using directly the frame-rate of 25 time steps per second, the location at time $t+1$ would be almost a Dirac mass on the location at the previous time step. Hence, we use a time discretization of 0.5s, which has the drawback of increasing the size of the neighborhood to consider for $P(L_{t+1} \,|\, L_t, M_t)$. In practice an individual can move up to 4 or 5 spatial locations away in one time step, which leads to a neighborhood of more than 50 locations.

The issue to face when choosing these probability models is the lack of training data. It would be impossible for instance to model these distributions exhaustively as histograms, since the total number of bins for $G \simeq 1,000$ and $M = 2$, if we consider transitions only to the 50 spatial neighbor locations and all possible maps, would be $\simeq 1,000 * 2 * 50 * 10 = 10^6$, hence requiring that order of number of observations. To cope with that difficulty, we interpolate these mappings with a Gaussian kernel from a limited number Q of control points, hence making a strong assumption of spatial regularity.

Finally, our motion model is totally parametrized by fixing the locations $l_1, \ldots, l_Q \in \{1, \ldots, G\}^Q$ of control points in the area of interest (where Q is a few tens), and for every point l_q and every map m by defining a distribution $\mu_{q,m}$ over the maps and a distribution $f_{q,m}$ over the locations.

From these distributions, for every map m and every location l, we interpolate the distributions at l from the distributions at the control points with a Gaussian kernel κ:

$$
\begin{aligned}
& P(L_{t+1} = l', M_{t+1} = m' \,|\, L_t = l, M_t = m) && (2)\\
& = P(L_{t+1} = l' \,|\, L_t = l, M_t = m) P(M_{t+1} = m' \,|\, L_t = l, M_t = m) && (3)\\
& = \left\{ \frac{\sum_q \kappa(l, l_q)\, f_{q,m}(l - l')}{\sum_r \kappa(l, l_r)} \right\} \left\{ \frac{\sum_q \kappa(l, l_q)\ \mu_{q,m}(m')}{\sum_r \kappa(l, l_r)} \right\} \ . && (4)
\end{aligned}
$$

Remains the precise definition of the motion distribution itself $f_{q,m}(\delta)$, for which we still have to face the scarcity of training data compared to the size of the neighborhood. We decompose the motion δ into a direction and a distance

and make an assumption of independence between those two components:

$$f_{q,m}(\delta) = P(L_{t+1} - L_t = \delta \,|\, L_t = l_q, M_t = m) \tag{5}$$
$$= g_{q,m}(\|\delta\|)\, h_{q,m}(\theta(\delta)) \ , \tag{6}$$

where $\|.\|$ denotes the standard Euclidean norm, g is a Gaussian density, θ is the angle quantized in eight values and h is a look-up table, so that $h(\theta(.))$ is an eight-bin histogram.

Finally, the complete parametrization of our model requires, for every control point and every map, M transition probabilities, the two parameters of g and the eight parameters of h, for a total of $Q * M * (M + 2 + 8)$ parameters.

3.2 Training

We present in this section the training procedure we use to estimate the parameters of the model described in the previous section. We denote by α the parameter vector of our model (of dimension $Q * M * (M + 2 + 8)$) and index all probabilities with it.

Provided with images from the video cameras, the ultimate goal would be to optimize the probability of the said sequence of images under a joint model of the image and the hidden trajectories, which we can factorize into the product of an appearance model (i.e. a posterior on the images, given the locations of individuals) with the motion model we are modeling here. However, such an optimization is intractable. Instead, we use an *ad-hoc* procedure based on the multi-camera multi-people tracking Probability Occupancy Map (POM) algorithm [2] to extract trajectory fragments and to optimize the motion model parameters to maximize the probability of those fragments.

Generating the Fragments. To produce the list of trajectory fragments we will use for the training of the motion model, we first apply the POM algorithm to every frame independently. This procedure optimizes the marginal probabilities of occupancy at every location in the area of interest so that a synthetic image produced according to these marginals matches the result of a background-subtraction pre-processing. We then threshold the resulting probabilities with a fixed threshold to produce finally at every time step t a small number N_t of locations $(l_1^t, \ldots, l_{N_t}^t) \in \{1, \ldots, G\}^{N_t}$ likely to be truly occupied.

To build the fragments of trajectories we process pairs of consecutive frames and pick the location pairing $\Xi \subset \{1, \ldots, N_t\} \times \{1, \ldots, N_{t+1}\}$ minimizing the total distance between paired locations $\sum_{\xi \in \Xi} ||l_{\xi_1}^t - l_{\xi_2}^{t+1}||$. If $N_t > N_{t+1}$, some points occupied at time t cannot be paired with a point at time $t + 1$, which corresponds to the end of a trajectory fragment. Reciprocally, if $N_t < N_{t+1}$, some points occupied at $t + 1$ are not connected to any currently considered fragment, and a new fragment is started.

We end up with a family of U fragments of trajectories

$$\mathbf{f}_u \in \{1, \ldots, G\}^{s_u}, \quad u = 1, \ldots, U \ . \tag{7}$$

E-M Learning. The overall strategy is an E-M procedure which maximizes alternatively the posterior distribution on maps of every point of every fragment $\mathbf{f}_u$, and the parameters of our motion distribution.

Specifically, let $\mathbf{f}_u^k$ denote the k-th point of fragment u in the list of fragments we actually observed. Let $\mathbf{F}_u^k$ and M_u^k denote respectively the location and the hidden map of the individual of fragment u at step k under our model.

Then, during the E step, we re-compute the posterior distribution of those variables under our model. For every first point of a fragment, we set it to the prior on maps. For every other point we have:

$$\nu_u^k(m) \tag{8}$$

$$= P_\alpha(M_u^k = m \,|\, \mathbf{F}_u^1 = \mathbf{f}_u^1, \ldots, \mathbf{F}_u^k = \mathbf{f}_u^k) \tag{9}$$

$$= \sum_{m'} P_\alpha(M_u^k = m \,|\, \mathbf{F}_u^{k-1} = \mathbf{f}_u^{k-1}, \mathbf{F}_u^k = \mathbf{f}_u^k,\ M_u^{k-1} = m')\, \nu_u^{k-1}(m') \tag{10}$$

$$\propto \sum_{m'} P_\alpha(\mathbf{F}_u^k = \mathbf{f}_u^k \,|\, \mathbf{F}_u^{k-1} = \mathbf{f}_u^{k-1},\ M_u^{k-1} = m') \cdot P(M_u^k = m \,|\, \mathbf{F}_u^{k-1} = \mathbf{f}_u^{k-1}, M_u^{k-1} = m')\, \nu_u^{k-1}(m') \tag{11}$$

$$= \sum_{m'} \left\{ \frac{\sum_q \kappa(\mathbf{f}_u^{k-1}, l_q)\, f_{q,m'}(\mathbf{f}_u^{k-1} - \mathbf{f}_u^k)}{\sum_r \kappa(\mathbf{f}_u^{k-1}, l_r)} \right\} \left\{ \frac{\sum_q \kappa(\mathbf{f}_u^{k-1}, l_q)\ \mu_{q,m'}(m)}{\sum_r \kappa(\mathbf{f}_u^{k-1}, l_r)} \right\} \nu_u^{k-1}(m') \ . \tag{12}$$

From this estimate, during the M step, we recompute the parameters of $\mu_{q,m}$ and $f_{q,m}$ for every control point l_q and every map m in a closed-form manner, since there are only histograms and Gaussian densities. Every sample $\mathbf{f}_u^k$ is weighted with the product of the posterior on the maps and the distance kernel weight $\nu_u^k(m)\, \kappa(\mathbf{f}_u^k, l_q)$.

4 Results

In this section, we first present the video sequences that we acquired to test our algorithm and describe the behavioral models we learned from them. We then demonstrate how they can be used both to improve the reconstruction of typical trajectories and to detect atypical ones.

4.1 Synthetic Data

The first step taken to validate the correct functioning of our algorithm was to test it against synthetic data. We generated synthetic probability occupancy maps of people moving along predefined paths. New people were created at the beginning of paths according to a Poisson distribution. Their speed followed a Gaussian distribution and their direction of movement was randomized around the paths. When two or more paths were connected, we defined transition probabilities between them, and people were switching paths accordingly.

The results on the synthetic data have been fully satisfying, as the retrieved behavioral maps correctly reflected the different paths we created.

4.2 Training Sequences

To test our algorithm, we acquired two multi-camera video sequences using 3 standard DV cameras. They were placed at the border of a rectangular area of interest in such a way as to maximize camera overlap, as illustrated by Fig. 1. The area of interest is flat and measures about 10m by 15m. The 3 video streams were acquired at 25 fps and later synchronized manually.

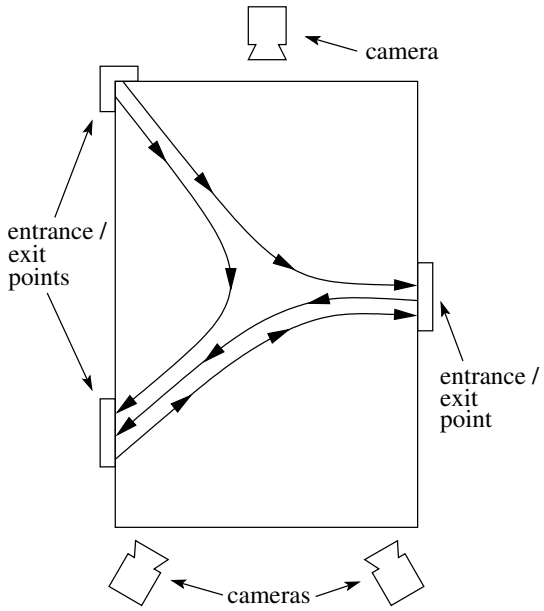


Fig. 1. Top view of the scenario used for algorithm training. People are going from one entrance point to an exit point using one of the available trajectories.

We use the first video, which lasts about 15 minutes, for training purposes. It shows four people walking in front of the cameras, following the predefined patterns of Fig. 1 that involve going from one entrance point to another.

In a second 8-minute-long test sequence, the same 4 people follow the same patterns as the training sequence for about 50 percent of the time and take randomly chosen trajectories for the rest. These random movements can include standing still for a while, going in and out of the area through non standard entrance points, taking one of the predefined trajectory backwards, etc. Screen shots of the test sequence with anomaly detection results are displayed on Fig. 7.

4.3 Behavior Model

As described in § 3.2, we first apply the POM algorithm [2] on the video streams, which yields ground plane detections that are used by our EM framework to construct the behavior model.

The ground plane of the training sequence is discretized into a regular grid of 30×45 locations. Probability distribution maps are built using one control point every 3 locations. The behavioral model of the 15 minute long training sequence is generated using 30 EM iterations, which takes less than 10 minutes on a 3 GHz PC using no particular optimization.

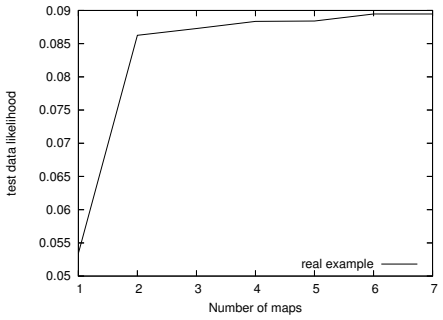
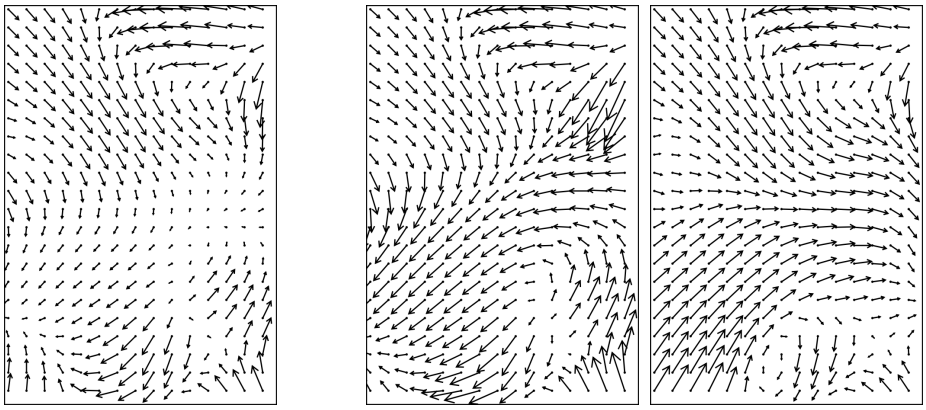


Fig. 2. Cross-validation: to find the ideal number of maps to model a given scenario, we run our learning algorithm with different number of maps on 80% of the training sequence. We then use the other 20% to compute the likelihood of the data given our model. In our training sequence, that is shown here, 2 maps are enough to model the situation correctly.

We use cross-validation to choose the number of maps that gives the most significant model. We apply our learning algorithm several times on 80% of the training sequence with each time a different number of maps, as shown on Fig. 2. The rest of the sequence is used to compute the likelihood of the trajectories under our model. In the end, we choose the smallest number of maps, which accurately captures the patterns of motion. On our testing sequence, it turns out that two maps are already sufficient. Figure 3 displays the behavioral maps that are learned in the one-map (left) and two-map (right) cases. By comparing them to Fig. 1, one can see that the two-map case is able to model all trajectories of the scenario.

Figure 4 shows the probabilities of staying in the same behavioral map over the next half second. These probabilities are relatively high, but not uniform over the whole ground plan, which indicates that people are more likely to switch between maps in some locations.



(a) One map (b) Two maps

Fig. 3. Motion maps in the top view resulting from the learning procedure, with one map (a) or two maps (b). The difficulty of modeling a mixture of trajectories under a strict Markovian assumption without an hidden state appears clearly at the center-right and lower-left of (a): Since the map has to account for motions in two directions, the resulting average motion is null, while in the two-map case on (b) two flows appear clearly.

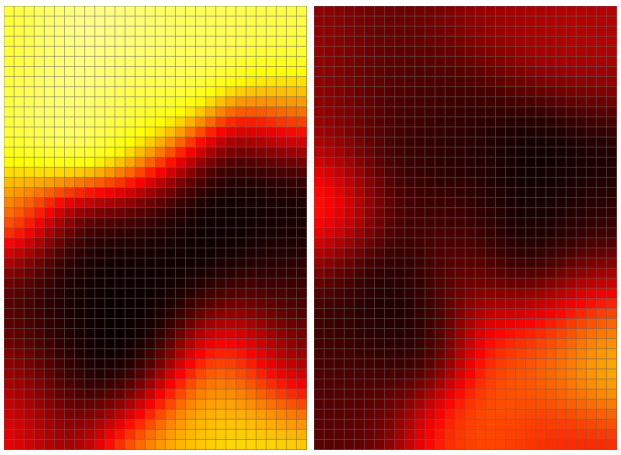


Fig. 4. Probability to remain in the map 0 (left) and in the map 1 (right) in the two-map case. Dark color indicates a high probability.

4.4 Tracking results

Here, we discuss the benefits of using behavioral maps learned with our algorithm to improve the performance of a people tracker. To this purpose, we have implemented the multi-people tracking algorithm of [2]. This work combines dynamic programming with a color model and an isotropic motion model to extract people trajectories. To integrate our behavioral maps, we have customized the tracking algorithm by replacing the uniform isotropic motion model with our model.

The behavioral maps had to be adapted to fit into the dynamic programming framework. Specifically, from every behavioral map, we generated a motion map that stores, for each position of the ground plane, the probability of moving into one of the adjacent position at the next time frame.

The main difference with [2] is that a hidden state in the HMM framework is now characterized by both a map and a position. Also the transition between HMM states is now given by both a transition probability between maps and between locations. The rest of the tracking framework, however, has been untouched.

To quantify the benefits of the behavioral maps, we started by running the original tracking algorithm [2] on our training sequence. We then ran our modified version on the same sequence, using in turn a one-map behavior model and a two-map one.

A ground truth used to evaluate the results was derived by manually marking the position and identity of each person present on the ground plane for every 10 frames. Scores for both algorithms were then computed by comparing their results to the ground truth. For this purpose, we define a trajectory as being the path taken by a person from the time it enters the area until it exits it. For every trajectory of the ground truth, we search if there is a matching set of detections from the algorithm results. A true positive is declared when, for every position of a ground truth trajectory, a detection is found within a given distance R, and all detections correspond to the same identity. If there is a change in identity, it obviously means that there has been a confusion between the identities of two people, which cannot be considered as a true positive. The false positive value is the average number of false detections per time frame. Results from Table 1 show both false positive and negative values for [2] and the modified algorithm using a one-map and a two-map behavior model. Results are shown for 3 different values of the distance R.

It appears from Table 1 that for about the same number of false positives, using 1, respectively 2, behavioral maps helps reducing significantly the number of false negatives. Moreover, one can notice that the paths are found with greater precision, when using two behavioral maps, since the number of false negatives is no longer influenced by the distance R.

4.5 Anomaly Detection Results

Detecting unlikely motions is another possible usage of the behavioral maps computed by our algorithm. We show the efficiency of this approach by applying it

Table 1. The false negative value corresponds to the number of trajectories out of a total of 75, that were either not found or were not consistent with the ground truth. The false positive value stands for the average number of false detections per time frame.

	$R = .5$m		$R = 1$m		$R = 2$m	
	FN	FP	FN	FP	FN	FP
Fleuret et al.	17	0.18	14	0.15	13	0.15
Our algorithm, one map	15	0.22	13	0.20	12	0.19
Our algorithm, two maps	10	0.21	10	0.18	10	0.17

for classifying trajectories from the test sequence into "normal" or "unexpected" category.

We start by creating a ground truth for the test sequence. We manually label each trajectory depending on whether it follows the scenario of Fig. 1 or not.

For every trajectory, a likelihood score is computed using the behavioral maps. For this we proceed using an HMM framework, in which our hidden state is the behavioral map the person is following. The transition between states is given by the transition probabilities between maps and the observation probability is the probability of a move, given the map the person is following. Having defined all this, the likelihood of a trajectory is simply computed using the classical forward-backward algorithm. The score is then compared to a threshold to classify the trajectory as "normal" or "unexpected".

We classified the 47 trajectories automatically retrieved from the test sequence using a one-map and a two-map behavior models. The results are displayed on Table. 2 and show the improvement when using several maps: the behavior model with only one map produces 7 (respectively 29) false positives if missing only one (respectively zero) abnormal trajectories, when the two-map models reduces this figure to 4 (respectively 9).

Table 2. Error rate for atypical trajectory detection. The total number of retrieved trajectories is 47, among which 16 are abnormal. With either one or two maps, the number of false positives (i.e. trajectories flagged as abnormal while they are not) drops to 1 for a number of false-negatives (i.e. non flagged abnormal trajectories) greater than 2. However, for very conservative thresholds (less than 2 false-negatives) the two-map model the advantage of using two maps appears clearly.

FN	FP	
	One map	Two maps
0	29	9
1	7	4
2	1	1

Instead of computing a score for a complete trajectory, one can also generate a score for a small part of it only, using the very same technique. This way of doing is more appropriate for monitoring trajectories in real time, for instance embedded in a tracking algorithm. This leads to a finer analysis of a trajectory, where only the unexpected parts of it are marked as such.

This procedure can be used directly to "tag" individuals on short time interval in the test video sequence. Figure 6 shows a selected set of atypical behavior, according to our two-map model. The unlikely parts of the trajectories are drawn using dotted-style lines. This should be compared to the two right maps of Fig.3. On the other hand, Fig. 5 shows some trajectories that follow the predefined scenario. Finally, Fig. 7 illustrates the same anomaly detection results, projected on camera views.

5 Conclusion

We have presented a novel model for motions of individuals which goes beyond the classical Markovian assumption by relying on a hidden behavioral state. While simple, this model can capture complex mixture of patterns in public places and can be trained in a fully unsupervised manner from video sequences.

Experiments show how it can be integrated in a fully automatic multi-camera tracking system and how it improves the accuracy and reliability of tracking in ambiguous situation. Moreover, it allows for the characterization of abnormal trajectories with a very high confidence.

Future work will consist of extending the model so that it can cope with a larger class of abnormal behaviors which can be characterized only by looking at statistical inconsistency over long period of time or inconsistency between the individual appearance and the motion.

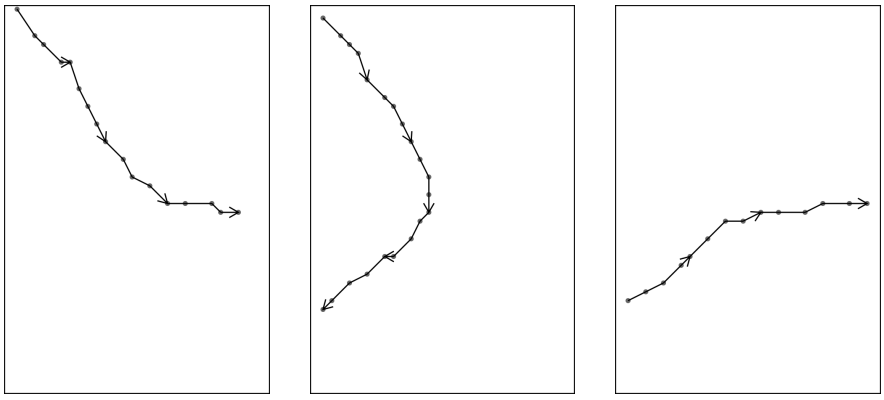
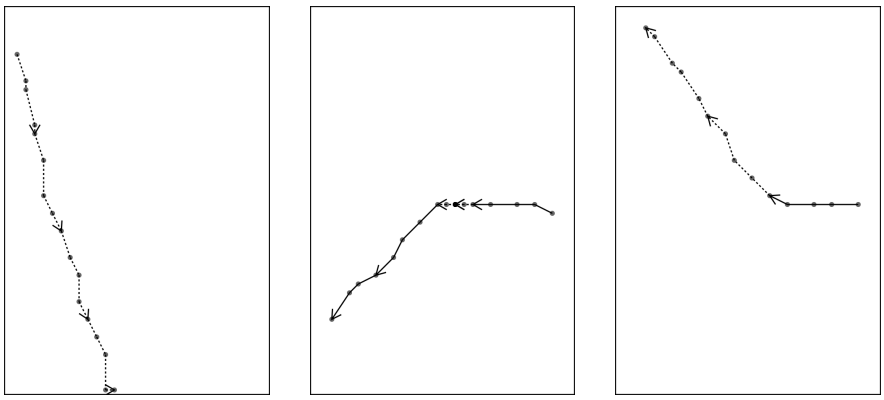


Fig. 5. Three example of retrieved "normal" trajectories, according to the scenario illustrated on Fig. 1.



(a) (b) (c)

Fig. 6. Three examples of atypical retrieved trajectories, according to the scenario illustrated on Fig. 1. Unlikely parts are displayed with dotted-style lines. a) The person is taking an unusual path; b) The person is stopping (middle of the trajectory); c) The person is taking a predefined path backward.

References

1. Khan, S., Shah, M.: A multiview approach to tracking people in crowded scenes using a planar homography constraint. In: European Conference on Computer Vision. Volume 4. (2006) 133–146
2. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-camera people tracking with a probabilistic occupancy map. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2) (2007) 267–282
3. Zhao, T., Nevatia, R.: Tracking multiple humans in crowded environment. In: Conference on Computer Vision and Pattern Recognition. (2004)
4. Smith, K., Gatica-Perez, D., Odobez, J.M.: Using particles to track varying numbers of interacting people. In: Conference on Computer Vision and Pattern Recognition. (2005)
5. Kang, J., Cohen, I., Medioni, G.: Tracking people in crowded scenes across multiple cameras. In: Asian Conference on Computer Vision. (2004)
6. Oh, S., Russell, S., Sastry, S.: Markov chain monte carlo data association for general multiple-target tracking problems. In: IEEE Conference on Decision and Control, Paradise Island, Bahamas (2004)
7. Bui, H., Venkatesh, S., West, G.: Policy recognition in the abstract hidden markov models. Journal of Artificial Intelligence Research **17** (2002) 451–499
8. Berclaz, J., Fleuret, F., Fua, P.: Pom: Probability occupancy map (2007) http://cvlab.epfl.ch/software/pom/index.php.
9. Johnson, N., Hogg, D.: Learning the distribution of object trajectories for event recognition. In: British Machine Vision Conference. (1995)

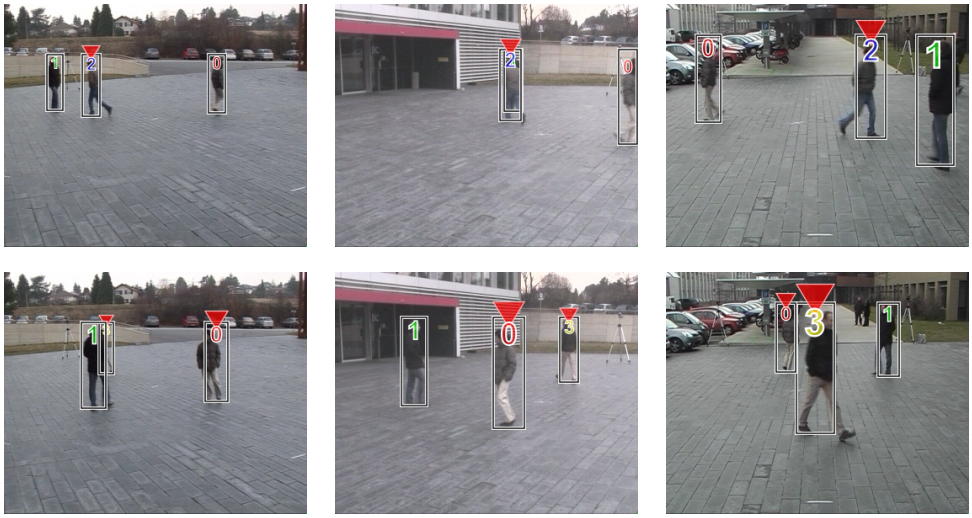


Fig. 7. Anomaly detection in camera views. Each row consists of views from three different cameras at the same time frame. A red triangle above a person indicates that it does not move according to the learned model.

10. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8) (August 2000) 747–757
11. Bennewitz, M., Burgard, W., Cielniak, G.: Utilizing learned motion patterns to robustly track persons. In: Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS). (2003)
12. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. Systems, Man, and Cybernetics, Part B, IEEE Transactions on **35**(3) (June 2005) 397–408
13. Hu, W., Xiao, X., Fu, Z., Xie, D., Tan, T., Maybank, S.: A system for learning statistical motion patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(9) (September 2006) 1450–1464
14. Antonini, G., Venegas, S., Thiran, J.P., Bierlaire, M.: A discrete choice pedestrian behavior model for pedestrian detection in visual tracking systems. In: Advanced Concepts for Intelligent Vision Systems. (2004)