Learning and regularizing motion models for enhancing particle filter-based target tracking

Francisco Madrigal, Mariano Rivera, and Jean-Bernard Hayet

Centro de Investigación en Matemáticas,
Jalisco s/n, Col. San Javier
36240 Guanajuato, GTO, México

Abstract. This paper describes an original strategy for using a data-driven probabilistic motion model into particle filter-based target tracking on video streams. Such a model is based on the local motion observed by the camera during a learning phase. Given that the initial, empirical distribution may be incomplete and noisy, we regularize it in a second phase. The hybrid discrete-continuous probabilistic motion model learned this way is then used as a sampling distribution in a particle filter framework for target tracking. We present promising results for this approach in some common datasets used as benchmarks for visual surveillance tracking algorithms.

1 Introduction

Visual target tracking has been the object of a very large amount of research in the last two decades, mainly in the communities of computer vision, image processing and networks. This is motivated in particular by a strong demand of automatic tracking tools for applications such as video-conferencing, gesture analysis, TV broadcasting, wildlife studies or video-surveillance, among many others. Our work targets particularly tracking in video-surveillance applications, which is characterized by a number of specific problems. First, tracking in that case is in general limited to pedestrians or cars. Moreover, a scene monitored by a surveillance camera typically contains many potential targets to track. They are not known in advance and have to be detected automatically. Furthermore, they generally undergo partial to complete occlusions, both from scene clutter (walls, pillars, poles...) or from other targets. Last, it is common in outdoors scenes that the appearance of different targets is quite similar, i.e. it may be hard to distinguish one target from another. In this difficult context and when dealing with only one camera, the motion model, i.e. the a priori knowledge about how objects move in the scene, helps to keep track of targets that are either ambiguous (if their appearance is similar to the one of a neighbour target) or occluded. The most simple and common motion models in the literature are constant velocity or constant acceleration ones, but they may not handle the specificities of a particular scenario: For example, because of the topology of one place, it may be frequent that pedestrians make sharp turns, which cannot be handled by simple motion models. Our contribution has been to infer a more complex probabilistic

motion model, by using the low-level information collected by the camera. To the best of our knowledge, this approach has not been proposed before, and the results we present in Section 5 show how promising it is. We will detail the construction of the prior in Section 3 and see how to use it in a tracking algorithm in Section 4. After presenting experimental results, we make a conclusion and give hints for future work in Section 6.

2 Related work

Much progress has been done recently in the literature for the design of efficient tracking techniques. If early works have essentially imported filtering techniques from the radar community (e.g., the Kalman filter), later works have heavily relied on probabilistic models for the target appearance, e.g. Meanshift and its variants [1], or color histogram-based models incorporated into particle filters [2]. The latest trend in the area of multiple target tracking is the paradigm of tracking-by-detection [3, 4], i.e. the coupling of tracking with powerful machine learning-based object detection techniques, that detect, by parts or as a whole, objects from some given class of interest (in particular, pedestrians [5]) with high confidence levels. The problem of tracking is then transformed into an optimization problem that finds a partition of space-time detections into coherent trajectories. The main limitation of these approaches is that their success depends essentially in the pedestrian detector. If it fails, the whole tracking system becomes inefficient. The problem is that in presence of occlusions, as it occurs frequently in our case, pedestrian detectors are likely to fail. Other detection-free techniques have been proposed [6] more recently, but they share with detection-based methods the need to wait for a given window of time before completing the association problem.

Here, we use a more traditional approach for tracking, namely the particle filter. We do not focus on the observation model, but rather on the other key element of any probabilistic tracker, the motion model. Works related to ours, i.e. based on developing new probabilistic motion models, is for example the one of [7], where the authors use a database of 3D motions to design a probabilistic model for tracking articulated objects. In this work, we aim at capturing the complexity of 2D motion in a given scene into a probability distribution. Such a complexity may be the consequence of specific physical elements (e.g., walls, corridors...) and it results in simple motion models being incorrect.

3 Learning motion transition models from sequences

This section describes our framework to learn motion priors from the low level information extracted from video sequences.

3.1 Estimating empirical state transition priors

The idea of our approach is to learn information on state transitions, where the state refers to the quantities that will be estimated during tracking, in our case

position and velocity. We will denote these quantities as $\mathbf{r} = (x, y)^T$ (position) and $\mathbf{v} = (v^m, v^\theta)^T$ (velocity, decomposed in a magnitude v^m and an orientation v^θ). From optical flow information computed in video sequences taken by the same camera that will do the tracking, we collect statistics about the *temporal* evolution of this state. It is done as compactly as possible, to make the handling of these distributions tractable. For example, if we refer to the time as t, we are interested in learning information about the joint distribution $p(\mathbf{v}_{t+1}, \mathbf{r}_{t+1}|\mathbf{v}_t, \mathbf{r}_t)$, that sums up the a priori knowledge about where and at what velocity targets tend to be at time $t + 1$, when their position and velocity at t are given. Modeling this joint distribution in the continuous domain would require parametric distributions (e.g. mixtures of Gaussian distributions) difficult to handle, e.g. in a regularization framework. Similarly, a fully discretized distribution would require a lot of memory. Hence, we adopted a hybrid representation, partly discrete, partly continuous. We factorize the probabilistic state transition model as follows, by supposing conditional independence between the two velocity components, given the previous state,

$$p(\mathbf{v}_{t+1}, \mathbf{r}_{t+1}|\mathbf{v}_t, \mathbf{r}_t) \approx p(v^m_{t+1}|\mathbf{v}_t, \mathbf{r}_t)p(v^\theta_{t+1}, \mathbf{r}_{t+1}|\mathbf{v}_t, \mathbf{r}_t). \quad (1)$$

For the first term, we adopt a simple, continuous Gaussian model, i.e. $v^m_{t+1} \sim \mathcal{N}(v^m_t, \sigma^2_m(\mathbf{v}_t, \mathbf{r}))$. For the second term, which is a priori more complex in nature, as it represents potential direction changes, we use a discrete distribution. We get estimates for these two distributions from optical flow data.

Collecting low-level information about state transitions. Our basic low-level data are point tracks given by a sparse optical flow algorithm. In our implementation, we used the Lucas-Kanade (LK) algorithm [8]. Note that this algorithm keeps track of only well-defined image points, i.e. points that are locally non-ambiguous, and that have two large eigenvalues in their autocorrelation matrix [9]. This makes the collected data a priori reliable as we do not take into account textureless areas or edge areas (in which the aperture effect applies); this would not be possible with a *dense* optical flow algorithm.

Furthermore, for the LK algorithm to be applied efficiently on long sequences, our implementation uses an image discretization into cells – see below – and refills each cell whenever the number of tracked points inside it goes below a given threshold. This avoids to refill globally the image with new corners and lose temporal state variations everywhere in the image at this moment.

Estimating the distribution of velocity amplitudes. Based on this sparse optical flow, we can first estimate the variance parameter $\sigma^2_m(\mathbf{v}_t, \mathbf{r})$ of $p(v^m_{t+1}, |\mathbf{v}_t, \mathbf{r}_t)$, by calculating the empirical variance of consecutive velocity magnitudes.

Estimating the distribution of velocity orientations. As for the second term of Eq. 1, we rely on a discrete representation of the $(v^\theta, \mathbf{r})$ space. We discretize the image into $D \times D$ pixels-wide cells (in our experiments, $D = 30$) and the set of possible orientations into other 8 cells. Then, we build, for each of these 3D

cells in the $(v_t^\theta, \mathbf{r}_t)$ space, a *discrete* representation of $p(v^\theta_{t+\tau(v^m)}, \mathbf{r}_{t+\tau(v^m)} | \mathbf{v}_t, \mathbf{r}_t)$. As space is discretized into cells, it is difficult to consider in that case motion information at time horizon 1, as most tracked points stay in the same cell; Instead, we collect temporal information, for each cell, at a time horizon that would span some significative motion towards a neighbour cell. A good choice for this time interval is $\tau(v^m) = \frac{D}{v^m}$, that represents the average time to cross the cell along a straight line. Hence, if a point is tracked by the LK algorithm, with velocity v^m at frame t, it contributes to the histogram at the 3D cell $(v_t^\theta, \mathbf{r}_t)$, into the bin (among $72 = 8 \times 9$ bins) corresponding to the observed $(v^\theta_{t+\tau(v^m)}, \mathbf{r}_{t+\tau(v^m)})$. This process is illustrated by Fig. 1, for a single point (in blue) tracked over a few frames. The central green dot gives a contribution for the cell representing its current state, based on its position $\tau(v^m)$ frames later. The corresponding bin is also depicted in green.

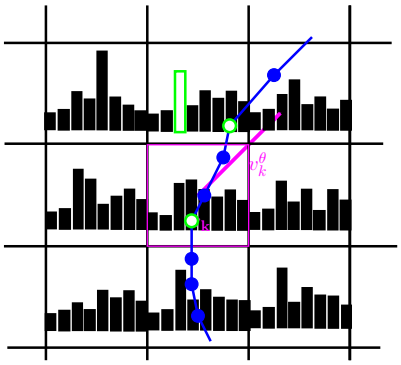

Fig. 1. Priors for the transition model by discretizing the image+orientations space. For one cell c_k, characterized by $(v_k^\theta, \mathbf{r}_k)$ (magenta), we model the distribution of $(v_{k'}^\theta, \mathbf{r}_{k'})$, i.e. the orientations and cells reached by the target after a "significant" amount of time $\tau(v^m)$. In green, one contribution from a point tracked on the blue path.

From now on, we will refer to each cell as $\mathbf{c}_k = (v_k^\theta, \mathbf{r}_k)$, with k indexing the set of cells, and $\mathbf{h}(\mathbf{c}_k)$ being the normalized histogram formed at that cell.

3.2 Regularization of velocity orientation distributions.

Let B be the number of bins in the histogram, $\mathbf{h}(\mathbf{c}_k)_i$ the value of the i-th bin in the normalized histogram, and $\log \mathbf{h}(\mathbf{c}_k)$ the histogram made of the logs of the entries of $\mathbf{h}(\mathbf{c}_k)$, i.e. $[\log \mathbf{h}(\mathbf{c}_k)]_i \stackrel{\text{def}}{=} \log [\mathbf{h}(\mathbf{c}_k)_i]$.

Now, as we will see later, given the high dimension of the space on which are defined all local distributions (72) and the relatively few data we have to estimate them, we need to estimate a regularized version $\mathbf{l}(\mathbf{c_k})$ of $\log \mathbf{h}(\mathbf{c}_k)$ with the optimization scheme described hereafter.$\mathbf{l}(\mathbf{c_k})$

The objective function we will minimize is the following one:

$$U(\mathbf{l}) = \frac{1}{2} \sum_{k \in K} \sum_{i,j} \mathbf{W}_{ij} ([\mathbf{l}_i(\mathbf{c}_k) - \log \mathbf{h}_j(\mathbf{c}_k)]^2 + \lambda \sum_{\mathbf{l} \in N(k)} [\mathbf{l}_i(\mathbf{c}_k) - \mathbf{l}_j(\mathbf{c}_l)]^2). \quad (2)$$

We can make several observations on this objective function. First, note that the searched optimum should reside in the manifold made of the concatenation of the logs of normalized histograms, i.e. for any cell $\mathbf{c}_k$, $\mathbf{l}(\mathbf{c}_k) \in \mathcal{L} = \{\mathbf{l} \in \mathbb{R}^B \text{ s.t. } \sum_{i=1}^{B} e^{\mathbf{l}_i} = 1\}$. Then, the first term of the function is a *data* term, that fits the $\mathbf{l}$'s to the empirical data. The second term is a smoothness constraint, that makes the histogram at one cell $\mathbf{c}_k$ similar to the cells $\mathbf{c}_l$ of its neighbourhood $N(k)$. These neighbourhoods include the spatial vicinity, of course, and the angular vicinity. Also note that λ is the regularization factor, and $K \subset \mathbb{N}$ is the set of considered cells. We will see below that we do not necessarily consider all the cells, but only the most informative ones.

Last, the terms $\mathbf{W}_{ij}$ are weights that encode the similarity between different, but close histograms bins, i.e. they soften the binning effect; they also include the particular fact that angle histograms are cyclic. For this purpose, we use the von Mises distribution with the angle between the direction vectors, which represent the bins i and j.

By developing a bit more the expression of $U(\mathbf{l})$ in Eq. 2,

$$U(\mathbf{l}) = C + \frac{1}{2} \sum_{\mathbf{k} \in K} (\mathbf{l}(\mathbf{c}_k)^T \mathbf{W}^{(\mathbf{1})} \log \mathbf{h}(\mathbf{c}_k) + \sum_{\mathbf{l} \in N(\mathbf{k})} \mathbf{l}(\mathbf{c}_k)^T \mathbf{W}^{(\mathbf{2})}(\mathbf{c}_k, \mathbf{c}_l) \mathbf{l}(\mathbf{c}_l)),$$

with C a constant, and

$$\mathbf{W}^{(1)}_{ij} = -2\lambda \mathbf{W}_{ij},$$

$$\mathbf{W}^{(2)}_{ij}(\mathbf{c}_k, \mathbf{c}_l) = \begin{cases} -2\lambda \mathbf{W}_{ij} & \text{if } k \neq l \\ \begin{cases} (1 + 2V\lambda) \sum_l \mathbf{W}_{il} - 2\lambda \mathbf{W}_{ii} & \text{if } i = j \\ -2\lambda \mathbf{W}_{ij} & \text{otherwise.} \end{cases} & \text{if } k = l. \end{cases}$$

Expressed this way, the problem of Eq. 2 is quadratical in the vector formed by all $\mathbf{l}$ and can be solved with classical numerical optimization techniques. We adopted a Gauss-Seidel scheme on the gradient of the objective function, in which after each iteration the new estimate for $\mathbf{l}$ is projected on the manifold $\mathcal{L} \subset \mathbb{R}^{|K|B}$ (i.e. re-normalization) [10]. In our experiments, approximately 40 iterations were necessary to ensure convergence, depending on the value chosen for the regularization parameter λ.

An example of regularized prior is shown (partially) in Fig. 2. It sums up the histograms $\mathbf{l}(\mathbf{c}_k)$ obtained with the optical flow data collected among several of the PETS'2009 video sequences, which is a dataset of common use in the

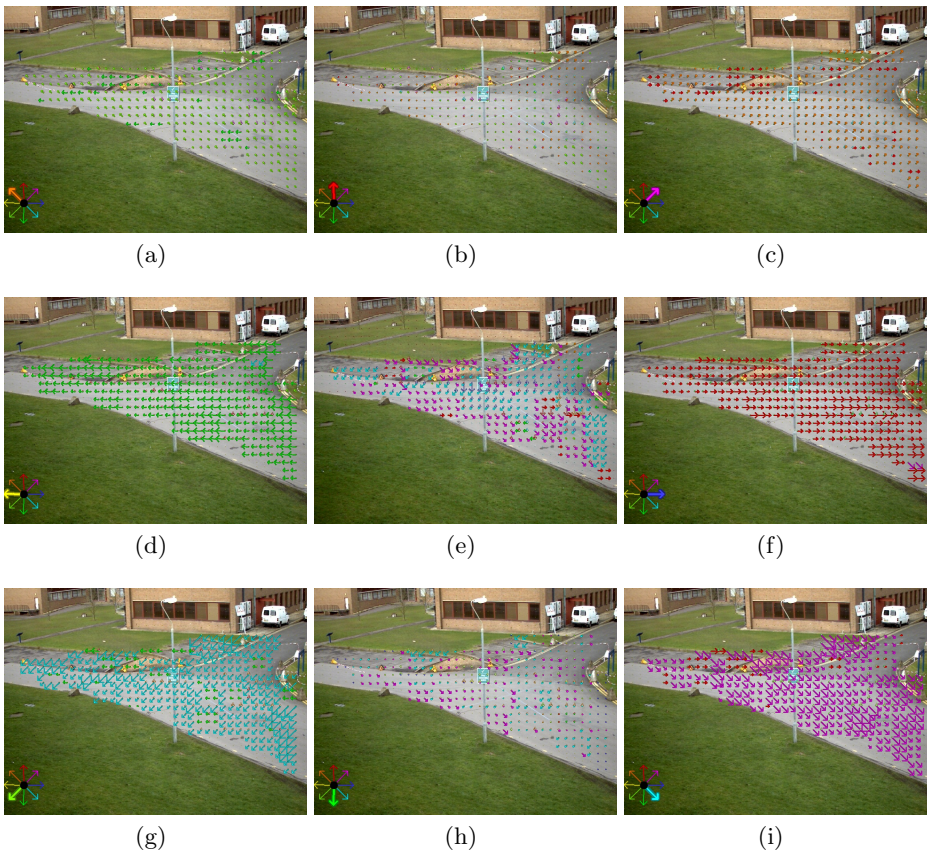

Fig. 2. A glimpse on the learned distribution $p(v^{\theta}_{t+1}, \mathbf{r}_{t+1}|\mathbf{v}_t, \mathbf{r}_t)$, for $v^{\theta}_t = \frac{\pi}{2}$ (downward motion, in white): each image corresponds to a spatial displacement (i.e., $\mathbf{r}_{t+1}$) indicated by the bold arrow on the bottom-left corner, and the colored arrows give the orientation (i.e., v^{θ}_{t+1}) with the highest bin among those corresponding to this $\mathbf{r}_{t+1}$.

evaluation of tracking algorithms [11]. The discretization being 3D, we depicted, for one particular initial orientation $v^{\theta}_t = \frac{\pi}{2}$ (i.e., a downward motion, in white), the main orientation observed at each of the potential arrival cell. Each image corresponds to a potential arrival cell (which one is indicated by the bold arrow in the bottom-left corner), and the colored arrow at each cell indicates the most frequent orientation for that cell, its length indicating the value of the corresponding bin. A remarkable element, justifying not to use too simple motion models, is that for this vertical orientation, there is nearly no support for the cell immediately downward (Fig. 2(h)). From this camera, vertical trajectories are seldom observed, downward velocities correspond generally to trajectories oriented to the south-west (Fig. 2(g)) or to the south-east (Fig. 2(i)). Finally, in Fig. 3, we show more synthetic views of the transition distributions, through

(a) (b)

Fig. 3. Raw and regularized transition fields for an upward initial direction ($v_t^\theta = -\frac{\pi}{2}$). We depict, before and after regularization, the 3D cell having the maximal bin value, with a first arrow pointing to $\mathbf{r}_{t+1}$ and a second arrow having orientation v_{t+1}^θ.

the sole maximal bins attained, for a given initial orientation, which is the upward direction in this case. The three arrows displayed correspond to the initial orientation, the direction to the attained new cell, and the final orientation. On the left side, we depict the non-regularized version, and on the right side its regularized counterpart. As in Fig. 2, one can note that vertical motions are generally not followed along straight line: They tend to bend either to the left (for example in the bottom-right corner) or to the right (upper-right corner). This reflects the physical constraints present in the scene for pedestrians.

To complete the regularization process, we have also tried to fill in incomplete information. As written before, because of the lack of data, some cells have no information at all, or nearly no information: For example, areas that are not accessible to pedestrians or areas hidden behind occluding clutter, such as the central pole at Fig. 3. Hence, we adopted a strategy to fill in these gaps: First we generate a binary image from data with not enough information and we decompose it into convex regions. We estimate the boundaries of two regions that are spatially close and sample a point in each region. The pair of points must have an Euclidean distance inferior to a threshold (i.e. we assume that points far away are not physically attainable). Then we sample an orientation at each point and interpolate positions/orientations by a cubic polynomial. Finally, we generate histogram entries according to the polynomial curvature. This strategy helps completing the missing information in some areas. We will see in Section 5, that it helps preserving the trajectories continuity.

4 Using the learned motion models during tracking

With this acquired knowledge about how pedestrians tend to move in the scene, we can now define a corresponding probabilistic motion model for performing

target tracking, i.e. not only the low-level tracking of a single point as LK does, but the one of a complex target. One of the most flexible techniques for performing such a tracking task is the particle filter (PF), as it allows to integrate in an elegant way a probabilistic model on what we should observe from the target in the current image (observation model), and a probabilistic model on what we know about how this targets moves (motion model). Above all, it makes no strong assumption about these probability distributions, the only constraints being, in the simplest form of PF, that one could evaluate the observation model at any state, and that one could sample the motion model from any state.

4.1 Particle filter-based visual tracking

Formally, we will index the targets to track with indices m, and associate one individual filter to each target. Any of these filters estimates the $4-D$ Markov state of its associated target at time t, $\mathbf{X}_t^m = (\mathbf{r}_t^m, \mathbf{v}_t^m)^T$ from the sequence of images $\mathbf{I}_1, \ldots, \mathbf{I}_t$. In practice, for the problem of tracking, the state contains the target Region of Interest position and its velocity. We will suppose, as other authors [12], that we have a rough knowledge of the position of the camera with respect to the scene, so that, at one possible position for the target in he image, the corresponding scale of the target bounding box is known. To do the target state estimation, the PF uses the recursive application of Bayes rule, that leads to

$$p(\mathbf{X}_t^m|\mathbf{I}_1,\ldots,\mathbf{I}_t) = p(\mathbf{I}_t|\mathbf{X}_t^m)\int_{\mathbf{X}_{t-1}^m} p(\mathbf{X}_t^m|\mathbf{X}_{t-1}^m)p(\mathbf{X}_{t-1}^m|\mathbf{I}_1,\ldots,\mathbf{I}_{t-1})d\mathbf{X}_{t-1}^m.$$

To get an approximate representation of this posterior from the previous equation, PF uses a Monte-Carlo approach, with a set of weighted samples (or "particles") from the posterior distribution, $\{(\mathbf{X}_t^{m,(n)}, \omega_t^{m,(n)})\}_n$, where n indexes the particles [13]. At each time t, these samples are generated from a proposal distribution $q(\mathbf{X}_t^m|\mathbf{X}_{t-1}^m, \mathbf{I}_t)$, and their weights are recursively updated so that they reflect how much the particles consecutive states evaluate under the posterior distribution. When, as we do here, the proposal distribution is precisely the probabilistic motion model, i.e.

$$q(\mathbf{X}_t^m|\mathbf{X}_{t-1}^m, \mathbf{I}_t) = p(\mathbf{X}_t^m|\mathbf{X}_{t-1}^m),$$

then the weight update rule is simply $\omega_t^{m,(n)} = p(\mathbf{I}_t|\mathbf{X}_t^{m,(n)})\omega_{t-1}^{m,(n)}$. A resampling step is applied whenever the number of significative particles (measured by $\frac{1}{\sum_n (\omega^{m,(n)})^2}$) becomes inferior to a threshold.

Here, we have used such a particle filter with a rather simple and common observation model, based on existing works on visual tracking [2]. It combines two main visible features of the target, its color distribution and its motion distribution along the video sequence. The first feature (color) is encoded through 3 H, S, V histograms defined in each of two sub-regions defined over the target

region of interest, the upper one and the lower one, for a total of six histograms. The choice of dividing the target region into two comes with the idea of associating some spatial information to the appearance model, in addition of the pure color data. As pedestrians clothes have generally quite different color distributions in their upper and lower parts, this observation model gives a much more discriminative power. The second feature (motion) is also encoded into a histogram, that contains the distribution of grey value differences between the current and the previous images in the sequence. For all of these histograms (color and motion), we store a reference histogram at the first frame where the target is detected, which is further updated along the video sequence with a simple exponential decay rule. As all of these features have the same form, the overall likelihood takes the form

$$p(\mathbf{I}_t|\mathbf{X}_t^m) \propto \prod_f \exp\left(-\frac{d^2(\mathbf{h}_f^m(\mathbf{X}_t^m), \mathbf{h}_f^{m*})}{2\sigma_f^2}\right),$$

where d is a distance between histograms (Bhattacharya distance), $\mathbf{h}_f^{m*}$ is the reference histogram for feature f (among 7 features) and $\mathbf{h}_f^m(\mathbf{X}_t^m)$ is the histogram computed at the target position $\mathbf{X}_t^m$. σ_f^2 the variance on the error on the histogram distance, which is a set of parameters for the algorithm.

4.2 Using the learned motion model

Now, given the learned motion model, we define a proposal definition based on it. We will refer to the underlying state transition model as $\pi(\mathbf{X}_{t+1}^m|\mathbf{X}_t^m)$. To draw samples from this distribution, conditionally to the state at t of the particle n, $\mathbf{X}_t^{m,(n)}$, we use the following steps:

1. Determine the 3D cell $\mathbf{c}_k = (v_k^\theta, \mathbf{r}_k)$ (position+velocity direction) the particle $\mathbf{X}_t^{m,(n)}$ corresponds to;
2. From the learnt distribution, $p(v_{k'}^\theta, \mathbf{r}_{k'}|v_k^\theta, \mathbf{r}_k)$, sample a cell $\mathbf{c}_{k'}$ where the tracked target could probably go after some time, with its future orientation;
3. Sample a velocity amplitude $v_{t+1}^{m,(n)}$ from $p(v_{t+1}^m, |\mathbf{v}_t^{m,(n)}, \mathbf{r}_t^{m,(n)})$;
4. From the pairs of cells, and their corresponding orientations, build a cubic curve joining the center cells and tangent to the initial and final orientations (i.e. Hermite interpolation);
5. Translate this curve on the 2D position of the particle $\mathbf{X}_t^{m,(n)}$ and sample a point on it around the position given by the velocity amplitude $v_{t+1}^{m,(n)}$. To make any neighbor configuration reachable from one initial configuration, we also add a lateral noise along the polynomial curve.

This way, we can sample from $\pi(\mathbf{X}_{t+1}^m|\mathbf{X}_t^m)$. Now, we may also define a more classical constant velocity model, that we will refer to as $p(\mathbf{X}_{t+1}^m|\mathbf{X}_t^m)$, and a mixture regulated by a parameter γ.

$$\pi'(\mathbf{X}_{t+1}^m|\mathbf{X}_t^m) = \gamma p(\mathbf{X}_{t+1}^m|\mathbf{X}_t^m) + (1-\gamma)\pi(\mathbf{X}_{t+1}^m|\mathbf{X}_t^m). \quad (3)$$

PETS'2009, $\gamma = \frac{1}{2}$			
SFDA	ATA	N-MODP	MOTP
0.40	0.42	0.51	0.51
0.36	0.39	0.49	0.49
0.39	0.45	0.50	0.51
0.40	0.46	0.52	0.52

PETS'2009, γ as quality			
SFDA	ATA	N-MODP	MOTP
0.40	0.42	0.51	0.51
0.38	0.43	0.50	0.50
0.41	0.47	0.52	0.52
0.43	0.48	0.54	0.53

CAVIAR, γ as quality			
SFDA	ATA	N-MODP	MOTP
0.10	0.10	0.23	0.58
0.14	0.12	0.37	0.62
0.14	0.13	0.40	0.68
0.15	0.13	0.40	0.70

Table 1. Performance evaluation of tracking. Results using PETS'2009 and CAVIAR datasets: First row (of each table) show the results for a classic SIR particle filter, with a constant velocity motion model. The last three rows are our results using motion prior with raw data, regularized data and with the strategy of filling incomplete data. Note : All results are the median value of 30 experiments. On the right, γ is taken variable, as the quality measure of the filter; on the left, it is fixed.

Through this mixture, we have evaluated several ways to use the prior: as a fixed mixture proposal ($\gamma = \frac{1}{2}$), or as a proposal to be used whenever the filter undergoes difficulties, e.g. because of occlusions. In that case, the prediction from the constant velocity model is made risky, since the state estimation is poor. In this last case, the coefficient γ weighting the two distributions is a quality measure evaluating the current estimation by the particle filter. One simple way to define it is as the average likelihood after the observation model is taken into account: $\sum_n p(\mathbf{I}_t|\mathbf{X}_t^{m,(n)})\omega_t^{m,(n)}$.

5 Experimental results

We evaluated our proposal on two public datasets: CAVIAR and PETS'2009. The first one has a ground truth but the second one has not, so we have manually generated one for the occasion. To evaluate our results quantitatively, we used a now standard methodology developed by the PETS community [11]. Tracking quality is not always easy to quantify, hence several metrics have been proposed, and we use four of them: (1) Normalized Multiple Object Detection Precision (N-MODP), which reflects the target detection rate and precision; (2) Multiple Object Tracking Precision (MOTP), that measures the tracks precision; (3) Sequence Frame Detection Accuracy (SFDA), and (4) Average Tracking Accuracy (ATA), which measures tracks precision but takes more into account the shortening of trajectories. These four indicators take values in the interval $[0, 1]$ (1 being for high quality).

The table 1 presents some results obtained for these indicators, on the two aforementioned datasets. In each case, the first row gives the indicator levels obtained with a classical SIR particle filter using a constant velocity motion mode. Then, the next three rows give results for the same SIR trackers using

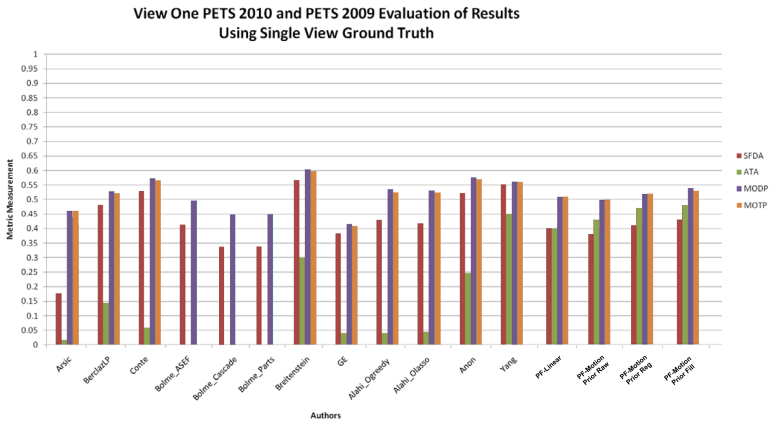

Fig. 4. Performance evaluation of tracking proposed by other authors and our proposal in set S2.L1 (view 1) of PETS 2009 dataset, for the four quality indicators. The last four results use our tracking system with linear motion model, motion prior model with raw data, a motion prior model with regularized data and a motion prior model with the strategy of filling zones with incomplete information. Other authors results have been reported in [11].

(1) the raw motion fields as probabilistic motion models; (2) the regularized motion fields and (3) the regularized motion fields with the hole filling strategy mentioned above. In all cases the results are improved, in particular for the first two indicators, that are sensible to the continuity of trajectories. Note that the results for the CAVIAR sequence are low, and this can be explained by the low sensibility of motion detection in that case (we used the OpenCV motion detector), that makes the tracker initialization long to occur, and makes the indicator levels drop in that case. Also note that we have compared two policies for the choice of γ, the policy setting it as a fixed value ($\gamma = \frac{1}{2}$, left column), and the policy of using the filter average likelihood as a measure. The best results are observed when taking the quality measure, i.e. when this proposal is really used when the filter is not tracking well the target. Last, Fig. 4 shows a comparison of our own results for the PETS'2009 sequence with results of other authors from PETS 2009 and 2010 conferences. As it can be noticed, the overall performance of our approach for the four indicators locates it among the best entries.

6 Conclusions and future work

We have presented a particle-based approach for visual tracking in video-surveillance sequences that relies, more than on a particularly efficient observation model, on a probabilistic motion model that is learned from sequences grabbed by the same camera. This way, the particle filter sampling is done in areas corresponding to paths that are much more likely to be taken by pedestrians

than what would be obtained with more traditional motion models (for example, the constant velocity motion model). Experiments on classical datasets of video-surveillance data and evaluation through standard indicators have shown that such an approach could be quite promising in obtaining better tracking results (where the term "better" may cover a complex reality, which is the reason why different tracking quality indicators have been used).

We plan several extensions for enhancing particle filter-based tracking: first, we plan to incorporate more scene-related elements in the regularization framework, i.e. to integrate boundaries (i.e. road limits) that should be taken into account, so as not to smooth motion fields through these boundaries; second, we plan to use this framework in an incremental way, so that no explicit learning phase would be required, and the model could be updated on a regular basis.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'00). Volume 2. (2000) 142–149
2. Perez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. Proc. of the IEEE **92**(3) (2004) 495– 513
3. Leibe, B., Schindler, K., Gool, L.V.: Coupled detection and trajectory estimation for multi-object tracking. In: Proc. Int. Conf. on Computer Vision. Proc. Int. Conf. on Computer Vision (2007)
4. Kuo, C.H., Huang, C., Nevatia, R.: Multi-target tracking by on-line learned discriminative appearance models. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'10). (June 2010) 685–692
5. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05). (2005) 878–885
6. Fragkiadaki, K., Shi, J.: Detection free tracking: Exploiting motion and topology for tracking and segmenting under entanglement. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'11). (2011)
7. Sidenbladh, H., Black, M.J., Sigal, L.: Implicit probabilistic models of human motion for synthesis and tracking. In: In Proc. of the European Conference on Computer Vision (ECCV'02). (2002) 784–800
8. Bouguet, J.: Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm. In: USENIX Technical Conference. (1999)
9. Shi, J., Tomasi, C.: Good features to track. In: Int. Conf. on Computer Vision and Pattern Recognition (CVPR'94). (1994) 593 – 600
10. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (2006)
11. Ellis, A., Ferryman, J.: Pets2010 and pets2009 evaluation of results using individual ground truth single views. Proc. of the IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS) (2010) 135 – 142
12. Madrigal, F., Hayet, J.: Multiple view, multiple target tracking with principal axis-based data association. In: Proc. of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance (AVSS). (2011)
13. Doucet, A., De Freitas, N., Gordon, N., eds.: Sequential Monte Carlo methods in practice. Springer (2001)