Multiple Target Tracking with Motion Priors

Francisco Madrigal Mariano Rivera Jean-Bernard Hayet

Centro de Investigación en Matemáticas (CIMAT)
Guanajuato, Gto., México
pacomd@cimat.mx, mrivera@cimat.mx, jbhayet@cimat.mx

Abstract. This paper presents a particle filter-based approach for multiple target tracking in video streams in single static cameras settings. We aim in particular to manage mid-dense crowds situations, where, although tracking is possible, it is made complicated by the presence of frequent occlusions among targets and with scene clutter. Moreover, the appearance of targets is sometimes very similar, which makes standard trackers often switch their target identity. Our contribution is two-fold: (1) we first propose an estimation scheme for motion priors in the camera field of view, that integrates sparse optical flow data and regularizes the corresponding discrete distribution fields on velocity directions and magnitudes; (2) we use these motion priors in a hybrid motion model for a particle filter tracker. Through several results on video-surveillance datasets, we show the pertinence of this approach.

1 Introduction

Visual target tracking has been the object of intense research in the last years, in a parallel trend to the huge development of visual surveillance (VS). If VS systems have invaded our daily lives, it is remarkable that many of its features still rely on human monitoring of up to dozens video screens. Computer vision techniques, such as motion detection and tracking, have started to be integrated in these systems, but we are still far to have a completely automatic system that would understand how people are moving, give high-level interpretation of the scene, or trigger alerts in case of suspect activities. The present work lies in this context of automatization of video surveillance systems, and in particular in the robust estimation of target trajectories, among mid-dense crowds. In that case, visual tracking is generally possible, but it is made difficult by frequent occlusions between targets and with scene clutter. Multiple, overlapping cameras settings may help to disambiguate among targets identities, but it is expensive, as more hardware is required. Here, instead, we focus on a cheap, *monocular* scheme and aim at tracking as many human targets as possible, in spite of the occlusions they may undergo. To reach that goal, we rely (1) on a probabilistic approach based on an observation model that can cope with partial occlusions, and (2) on a representation of the prior distributions of targets velocities, integrated as a proposal distribution in a particle filtering scheme. This second element, the use of motion prior, is the main contribution of this work, and to our knowledge it had not been proposed before in this form.

The organization of this paper is as follows: first, in Section 2, we give a short glance on the – huge – existing literature of monocular target tracking; we explain in Section 3 how velocities distribution are estimated, and in Section 4, we propose a particle filter-based scheme that uses these priors as a proposal distribution and motion model. Finally, in Section 5, we present experimental results on standard datasets of video-surveillance and in Section 6, we balance the pros and cons of our approach.

2 Related work

Target tracking in monocular video streams has generated a great amount of approaches, so that making an overview of the literature is a difficult exercise. Most early works have focused on coping with binary blobs detected from a background modeling algorithm [1], and with using stochastic processes tools imported from the radar community, i.e. by considering objects in motion as undistinguishable "dots". Then standard approaches have made extensive use of object appearance, such as color, geometric moments, spectral properties or, simply, shape.

Seminal appearance-based works include in particular the one of Comaniciu [2], who proposed the concept of Mean Shift tracking, i.e. of tracking the modes of the object likelihood, given its appearance. After the work of [3], in which *particle filters* were introduced for the first time in computer vision, in the context of active contours tracking, Monte-Carlo methods have taken the lead in the literature. Pérez et al. [4], on the one hand, and Nummiaro et al. [5], on the other hand, have proposed the first particle filter trackers based on color histograms, in the context of face tracking. It has proven to be particularly robust, e.g. in video-conferencing applications. The idea, which is still used in this paper, is to use a probabilistic observation model based on the color content of the target to track. Many more extensions have been proposed later on to the original principle of Sequential Monte-Carlo techniques, based on the combination of sampling strategies, probabilistic observation models and probabilistic motion models. On sampling mechanisms (which will be recorded briefly in this paper), novel ideas have been introduced from the statistics community [6], or from the machine learning community [7], among the others. On probabilistic observation models, many works have shown that particle filtering was flexible enough to integrate image-based information of very different nature [8]. On probabilistic motion models, on the contrary, few novel models have been proposed to enhance tracking algorithms, and most systems rely on very simple ones (constant velocity or acceleration, for example). Our work focuses on this part, and studies the use of motion priors learnt from the video monitoring of a scene to improve the motion model.

3 Motion priors

We build a regularized representation of the distribution of image velocities $\mathbf{v} = (v^m, v^\theta)$ in a monitored scene, where v^m is the velocity magnitude and v^θ its direction, given the position $\mathbf{r} = [x, y]$ in the image, i.e.

$$p(\mathbf{v}|\mathbf{r}) = p(v^\theta|\mathbf{r})p(v^m|\mathbf{r}, v^\theta). \qquad (1)$$

We model $p(v^m|\mathbf{r}, v^\theta)$ as a Gaussian distribution, as, when considering one motion direction at one point, people tend to walk at similar velocity magnitudes. We use a discrete distribution for the more complex $p(v^\theta|\mathbf{r})$. For estimating both distributions, we use motion data samples from video-sequences recorded in the same conditions as encountered during tracking, except that we use a much rougher information than precise, individual target tracking, namely optical flow. The optical flow algorithm to use can be either dense [9] or sparse [10]. In both cases, recollected velocity orientations are integrated into motion orientation histograms, whereas magnitudes are used to estimate the Gaussian conditionals.

From optical flow to velocity orientation histograms. For each frame t of the video sequence, we are given a motion field $\mathbf{v}_t(\mathbf{r}) = (v_t^m(\mathbf{r}), v_t^\theta(\mathbf{r}))$, computed by an optical flow algorithm. Histograms $\mathbf{h}(\mathbf{r})$ of instantaneous velocity directions at an image point $\mathbf{r} = [x, y]$ are incremented at the bin corresponding to the observed orientation $v_t^\theta(\mathbf{r})$, by the quantity

$$v_t^m(\mathbf{r})/(\kappa + v_t^m(\mathbf{r})), \qquad (2)$$

where κ serves as a velocity magnitude threshold, in order for static pixels (e.g. from the background) not to be taken into account. Moreover, to limit noisy contributions due to aperture effect, we exclude points for which the minimal value of the eigenvalue of the auto-correlation matrix is inferior to a threshold, as described in [11]. This avoids accumulating incorrect orientations from textureless moving areas.

Regularization of velocity orientation distributions. Let $\mathbf{h}(\mathbf{r})$ be the normalized histogram of instantaneous velocity directions observed in the video-sequence at an image point $\mathbf{r} = [x, y]$, belonging to some image region Ω. Let B be the number of bins in the histogram, and $\mathbf{h}(\mathbf{r})_i$ the value of the i-th bin in this normalized histogram. We define $\log \mathbf{h}(\mathbf{r})$ as the histogram made of the logs of the entries of $\mathbf{h}(\mathbf{r})$, i.e. $[\log \mathbf{h}(\mathbf{r})]_i \overset{\text{def}}{=} \log [\mathbf{h}(\mathbf{r})_i]$.

We estimate a smooth version $\mathbf{d}$ of the log of the histogram field, through the following optimization scheme in the manifold $\mathcal{D} = \{\mathbf{d} \in \mathbb{R}^B \text{ s.t. } \sum_i e^{\mathbf{d}_i} = 1\}$:

$$\min_{\mathbf{d} \in \mathcal{D}} U(\mathbf{d}) = \frac{1}{2} \sum_{\mathbf{r} \in \Omega} \sum_{i,j} \mathbf{W}_{ij}([\mathbf{d}_i(\mathbf{r}) - \log \mathbf{h}_j(\mathbf{r})]^2 + \lambda \sum_{\mathbf{s} \in N(\mathbf{r})} [\mathbf{d}_i(\mathbf{r}) - \mathbf{d}_j(\mathbf{s})]^2). \quad (3)$$

The first term is the *data* term, fitting the $\mathbf{d}$'s to the collected data. The second one is a smoothness constraint, that makes the histogram at one point $\mathbf{r}$

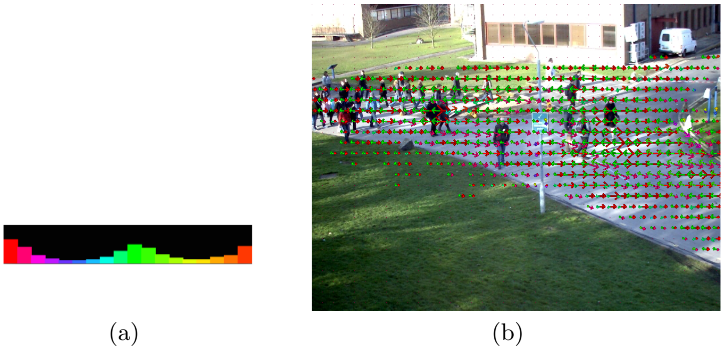

(a) (b)

Fig. 1. Velocity map. (a) Velocity orientations histogram for pixel $(560, 220)$. (b) The highest two local maxima in the velocity orientations histogram, for each 20th pixel.

similar to the ones of its neighbourhood $N(\mathbf{r})$. Terms $\mathbf{W}_{ij}$ encodes the similarity between some different, but close histograms bins, i.e. if the histograms are a bit shifted, they are somewhat similar; moreover, it includes the particular fact of our histograms to be angle histograms, i.e. for which bins are cyclic.

By developing the expression of $U(\mathbf{d})$, and after regrouping terms,

$$U(\mathbf{d}) = C + \frac{1}{2} \sum_{\mathbf{r} \in \Omega} (\mathbf{d}(\mathbf{r})^T \mathbf{W}^{(\mathbf{1})} \log \mathbf{h}(\mathbf{r}) + \sum_{\mathbf{s} \in N(\mathbf{r})} \mathbf{d}(\mathbf{r})^T \mathbf{W}^{(\mathbf{2})} \mathbf{d}(\mathbf{s})), \tag{4}$$

with C a constant, and

$$\mathbf{W}^{(1)}_{ij} = -2\lambda \mathbf{W}_{ij}, \tag{5}$$

$$\mathbf{W}^{(2)}_{ij} = \begin{cases} -2\lambda \mathbf{W}_{ij} & \text{if } \mathbf{s} \neq \mathbf{p} \\ \begin{cases} (1 + 2V\lambda) \sum_l \mathbf{W}_{il} - 2\lambda \mathbf{W}_{ii} & \text{if } i = j \\ -2\lambda \mathbf{W}_{ij} & \text{otherwise.} \end{cases} & \text{if } \mathbf{s} = \mathbf{p}. \end{cases} \tag{6}$$

Expressed this way, the problem of Eq. 3 can now be solved with classical optimization approaches. Here we adopted a Gauss-Seidel scheme in $\mathbb{R}^B$, in which after each iteration the new estimate for $\mathbf{d}$ is projected on the manifold $\mathcal{D}$ (i.e. re-normalization). In our experiments, some 40 to 60 iterations were necessary to ensure convergence, depending on the chosen value for λ (we set it to 0.5 in ours experiments).

Last, we used the regularized field to fill in image areas where not enough information was available. Typically, this occurs in zones where physical occlusions occur: an example in the datasets we used (see Fig. 2) is the pole at the center of the image. As most people pass behind it, no information is recollected here. However, we can infer the velocity fields in these regions as follows:

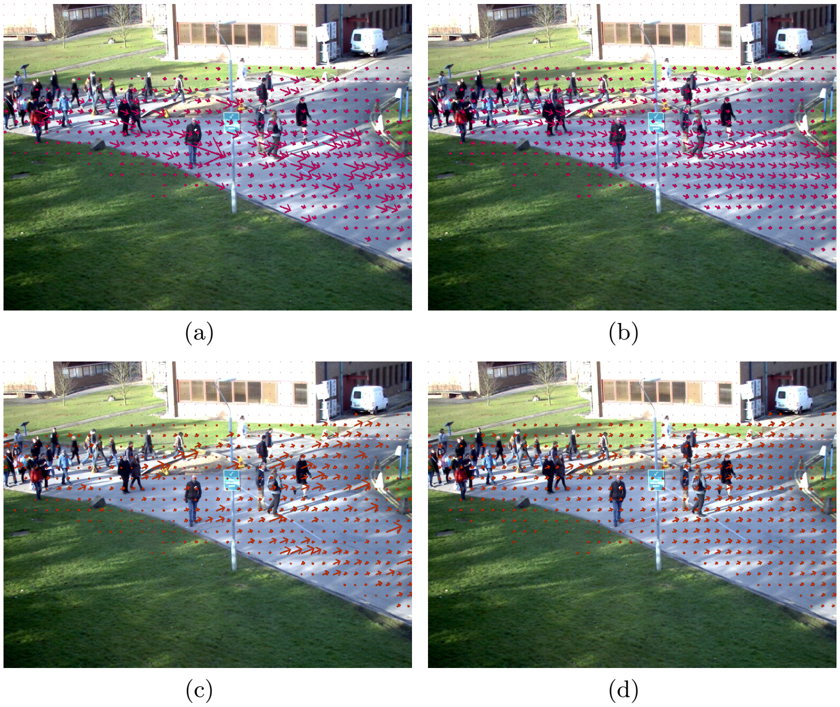


Fig. 2. Velocity orientation histogram fields, for two directions i (below and above). (a) and (c) Value in the raw histograms ($\mathbf{v}_k$). (b) and (d) Regularized histograms after convergence ($e^{\mathbf{d}_i}$). For clarity, data are shown every 20th pixel.

1. Form a binary image with data with not enough information and decompose it into convex regions.
2. Get the median axis of each shape and sample pairs of points on the borders, on each side of the median axis.
3. Sample orientations on each of the sampled point pairs.
4. Interpolate between these two positions/orientations by a cubic polynomial and generate histogram entries whenever the polynomial curvature stays below a threshold.

With the process above, we can fill in holes in the velocity field, and, as we will show in Section 5, this helps preserving the trajectories continuity.

Results of the histogram calculation and regularization are shown in Fig. 1. We computed the regularized velocity orientation map using > 3000-frames from different sequences of the PETS'2009 dataset [12]. The left image is an example of a histogram for a single pixel, while the right image shows the main two local maxima for the orientation histogram, at each pixel. To speed up the process, we also discarded from the process image regions where no significative motion was observed. In Fig. 1, these are the regions where no arrow is displayed, its complement being Ω. In Fig. 2, we show the regularization results for a particular direction (the one of the arrows), and the lengths of each arrow represent the value of the bin in the histogram. On the left side are the raw histograms, and we can appreciate how, because of the sparseness of the input data in some areas, we may have quite discontinuous orientation distributions around neighbour pixels. For this reason, regularization is critical to provide a smoother, more consistent map.

4 Tracking with particle filters and motion priors

This section details our particle filter-based tracker and the way it takes benefit from the motion priors computed beforehand.

4.1 Trackers: definition

As done classically, trackers are defined here as *stochastic filters* that estimate a state relative to the objects of interest, observed through the image. This state includes all the information about the targets needed to predict their posterior state. This could involve their position in the image (i.e. their bounding box center $[x, y]$), their apparent size ($[h, w]$) or their velocity ($[v^m, v^\theta]$). However, as we use particle filtering to perform the inference, it is important to limit the dimensionality of state vectors, to avoid an escalation in the number of required particles. Hence, we opted to set $[h, w]$ as a *deterministic* function of the position $[x, y]$, which is possible when a partial knowledge of the scene is available, as explained for example in [13]. In a few words, to estimate this mapping from $[x, y]$ to target sizes $[h, w]$, we make the assumptions (1) that the objects of interest are pedestrians and (2) that the observed scene is planar. We also suppose that the camera-scene geometry is roughly known, so that an estimate of the projection matrix is available. Hence, the state for a single target is reduced to the target position and its velocity, $\mathbf{X}_t = [x_t, y_t, v_t^m, v_t^\theta]^T$.

To detect new targets and initialize the corresponding trackers, we use a per-pixel statistical background model from a background modeling algorithm [1]. The most likely foreground pixels are grouped into connected components of a binary image (blobs). To manage exclusively pedestrians, we use several heuristics, e.g. a threshold on the size of the connected component (proportionally to the above position-to-scale mapping). We also avoid to validate new targets close to the predicted position of an existing tracker. If the blobs may correspond to a group of people, we use the heuristic of [14], based on the profile of the upper part of the binary region to separate it into individuals. After these filtering steps, we initialize a pedestrian tracker. As for the removal of trackers, e.g. when the corresponding target leaves the camera field of view, we maintain a quality coefficient γ (defined in the next sub-section) for each filter. Whenever γ stays below a threshold for a given period of time, we remove the tracker from the pool of trackers.

4.2 Tracking pedestrians with particle filters

Our trackers are implemented as variants of sequential importance resampling *particle filters* [15]. They require three main elements: a probabilistic observation model, a probabilistic motion model, and a proposal distribution. We explain the three of them in the following.

Observation model. As in [8], the targets appearance model uses color and motion. For each object k to track, we define and update color and motion reference histograms, so that in all frames, we can evaluate the likelihood of a possible target state $\mathbf{X}_{t,k}$ by comparing the histograms $\mathbf{h}_k^f$, computed at $\mathbf{X}_{t,k}$ for feature f, with the reference one, $\mathbf{h}_k^{f*}$. We use seven features in total: color histograms in the H, S, V channels on the upper and lower regions of the target[1] (6 features) and a motion histogram (1 feature). This motion histogram indexes absolute differences between consecutive images.

For each feature f, the comparison with the reference histogram relies on the Bhattacharya distance, referred to as D, between the reference histogram $\mathbf{h}_k^{f*}$ and the current one $\mathbf{h}_k^f(X_{t,k})$. The corresponding likelihood is defined as

$$P(\mathbf{Z}_t|\mathbf{X}_{t,k}) \propto \prod_f \exp\left(-\frac{D^2(\mathbf{h}_k^f(X_{t,k}), \mathbf{h}_k^{f*})}{2(\sigma^f)^2}\right), \tag{7}$$

where σ^f is the expected standard deviation on the Bhattacharya distance for feature f. Setting a large value on the histograms related to the V channel allows for example more robustness to illumination changes. Note that reference histograms are *updated* regularly in order to cope with changes in the image acquisition process (see for example [8] for details on the update mechanism).

Motion model(s). The probabilistic motion model encapsulates the a priori knowledge on how targets move $p(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$. A very simple and common motion model is a constant velocity model

$$\mathbf{X}_{t+1,k} = \begin{bmatrix} \mathbf{I}_{2\times2} & \mathbf{I}_{2\times2} \\ 0 & \mathbf{I}_{2\times2} \end{bmatrix} \mathbf{X}_{t,k} + S_{x,y}(\bar{\mathbf{X}}_{t,k})\nu_A, \tag{8}$$

where $\nu_A \sim N(\mathbf{0}, \mathbf{\Sigma}_A)$ is a zero-mean Gaussian additive noise with variance $\mathbf{\Sigma}_A$ and $S_{x,y}(\bar{\mathbf{X}}_{t,k})$ is a scale factor deduced from the mapping from $[x, y]$ to target size, described in 4.1. We will refer to this first model as $p^{(1)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$.

A second model is defined with the prior on velocities (Section 3), as

$$p^{(2)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k}) = \gamma p^{(1)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k}) + (1-\gamma)p^{(v)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k}), \tag{9}$$

where $p^{(1)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$ is the motion model in Eq. 8, and $p^{(v)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$ a model relying on the estimated velocity prior:

[1] The use of several histograms on different sub-regions is a common method to have some spatial information taken into account.

$$\mathbf{X}_{t+1,k} = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{I}_{2\times 2} \\ 0 & 0 \end{bmatrix} \mathbf{X}_{t,k} + \begin{bmatrix} 0 \\ \mathbf{v}^{(p)}_{t+1,k} \end{bmatrix}, \tag{10}$$

where $\mathbf{v}^{(p)}_{t+1,k} \sim p(\mathbf{v}_{t+1,k}|\mathbf{r}_{t+1,k})$. The idea is to use this prior as a proposal whenever the filter undergoes difficulties, e.g. because of occlusions, which makes the prediction from the constant velocity model risky, since the state estimation is poor. Hence, the coefficient γ weighting the two distributions is precisely the aforementioned quality measure evaluating the current estimation.

Proposal distribution. The third fundamental brick of the particle filter is the proposal distribution $q(\mathbf{X}_{t+1,k}|\mathbf{X}_{1:t,k}, \mathbf{Z}_{1:t})$, i.e. the distribution from which samples are generated at each step from the previous one. We chose its "bootstrap" form, where the proposal distribution is the same motion model, i.e. $q(\mathbf{X}_{t+1,k}|\mathbf{X}_{1:t,k}, \mathbf{Z}_{1:t}) = p(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$.

The particle filter in action. Given the previous three ingredients, the particle filter tracker is extremely simple to implement. It maintains a set of weighted particles $\{(\mathbf{X}^{(n)}_{t,k}, \omega^{(n)}_{t,k})_n\}$ and iterate the following steps

1. From the previous set of particles, generate a new set of particles from the proposal distribution, i.e. for all n

$$\mathbf{X}^{(n)}_{t+1,k} \sim q(\mathbf{X}_{t+1,k}|\mathbf{X}^{(n)}_{1:t,k}, \mathbf{Z}_{1:t+1}); \tag{11}$$

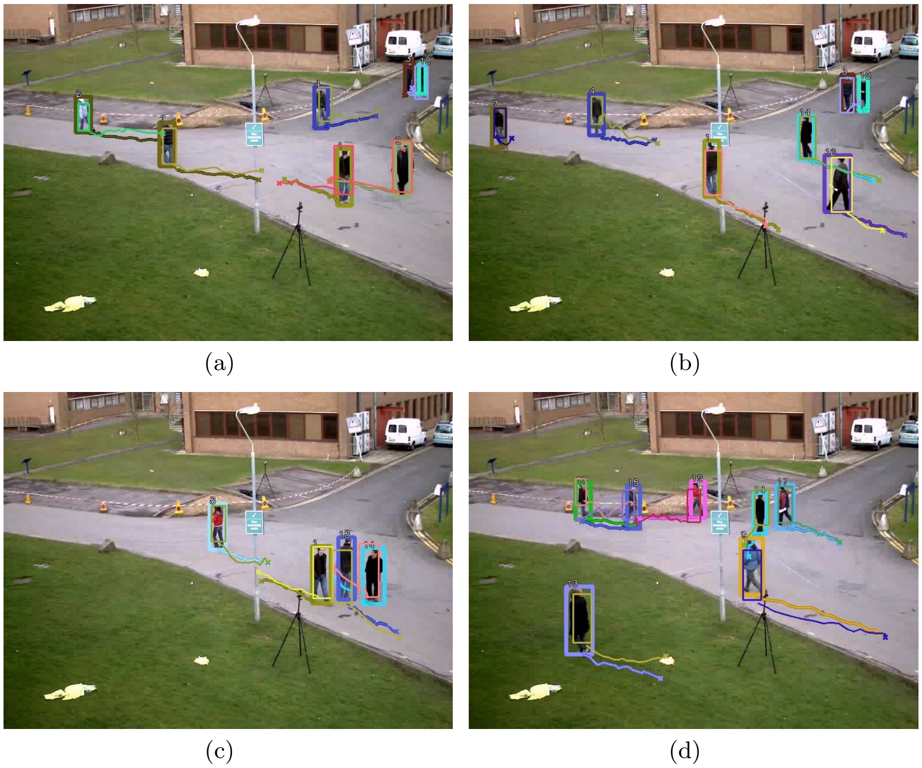
   In our case, we will evaluate all the motion models described above as proposal distributions (i.e. $p^{(1)}, p^{(2)}, p^{(3)}$);
2. Update particle weights with the particle state likelihood, the motion model and the proposal, i.e. for all n:

$$\omega^{(n)}_{t+1,k} \propto \frac{p(\mathbf{Z}_{t+1}|\mathbf{X}^{(n)}_{t+1,k})p(\mathbf{X}^{(n)}_{t+1,k}|\mathbf{X}^{(n)}_{t,k})}{q(\mathbf{X}^{(n)}_{t+1,k}|\mathbf{X}^{(n)}_{1:t,k}, \mathbf{Z}_{1:t+1})}\omega^{(n)}_{t,k} = p(\mathbf{Z}_{t+1}|\mathbf{X}^{(n)}_{t+1,k})\omega^{(n)}_{t,k}; \tag{12}$$

3. Compute a quality measure associated to the tracker as the average likelihood among particles.

$$\gamma_k = \sum_n p(\mathbf{Z}_{t+1}|\mathbf{X}^{(n)}_{t+1,k})\omega^{(n)}_{t,k}; \tag{13}$$

4. Normalize weights and compute the average state; test the weights variance, approximately $\sum_n [\omega^{(n)}_{t+1,k}]^2$; if it is superior to some threshold, the particle set is degenerate; then proceed to re-sample the particles with replacement.



(a) (b)

(c) (d)

Fig. 3. A few frames of tracking in sequence S2.L1. The tracks as estimated by our tracker and the ground truth tracks are superimposed.

5 Experimental results

We present some results of using our approach in real tracking situations. We employed public datasets such as PETS'2009. Since ground truth data is not available, we have manually generated one considering a rectangle around each target in the sequence. The two flavours of the particle filter we propose here (that differ by using or not the motion priors in the probabilistic motion model) are evaluated with a common methodology that has been used to evaluate the trackers performance [12]. Moreover, we also evaluate how the process of filling incomplete prior data improves tracking results.

Illustrations on multiple target tracking with motion prior are presented in Fig. 3. In the represented frames, we draw the feet position of the tracked pedestrian in the 20 previous frames. Thicker lines represent the ground-truth (GT) and thinner lines are our tracking estimation (TE). As it can be noticed, both are very close.

To evaluate our tracking results quantitatively, we used several metrics from the aforementioned evaluation scheme: (1) Normalized Multiple Object Detec-

Method	SFDA	ATA	N-MODP	MOTP
PF - Linear	0.40	0.42	0.51	0.51
PF - Motion Prior	0.42	0.46	0.51	0.52
PF - Motion Prior Fill	0.45	0.46	0.54	0.54

Table 1. Results for the S2.L1 sequence. The first row shows the result with different quality indicators for a classic SIR particle filter. The third row shows the results taking into account a motion prior and the last tracking results are obtained with motion priors and the strategy mentioned above of filling zones with incomplete information. Note: All results are the median value of 30 experiments.

tion 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 between 0 and 1 (1 being for the best tracker).

In table 1, we evaluate our tracking system with those metrics. Each row represent the average results obtained in different setups. The first one uses a traditional SIR particle filter tracker with a linear motion model. The second one is a SIR particle filter using the motion prior in the proposal. And the final experiment considers a motion prior with the strategy explained above of filling zones with incomplete information. When comparing the three rows, one can observe that N-MODP and MOTP have similar values but that SFDA and ATA increases substantially when incorporating motion priors. This means that the estimated trajectories, if not more precise, are longer, i.e we can keep the ID of a given pedestrian on longer periods of time. This is what we expected from using priors: in presence of occlusion from other targets or clutter (in which case the estimation of the velocity may be quite noisy) or in situations where the target undergoes non linear motion (i.e. in a zone where pedestrians tend to make sharp turns), the prediction step of the particle filter using motion priors sample particles in a way much closer to what people tend to do at this place than what would generate a too simple or too noisy motion model.

Last, in Fig. 4, we have compared the results of our own tracking strategies with the results obtained by several other authors as reported in [12]. For all indicators, the obtained results are quite competitive, and as we can see the most important result is the ATA metric (green column) which reflect the continuity of trajectories, in other words we can follow all targets using less trackers as other approaches.

6 Conclusion

We have described an original enhancement to color and motion-based particle filer tracking that relies on the use of motion priors. We showed how to estimate

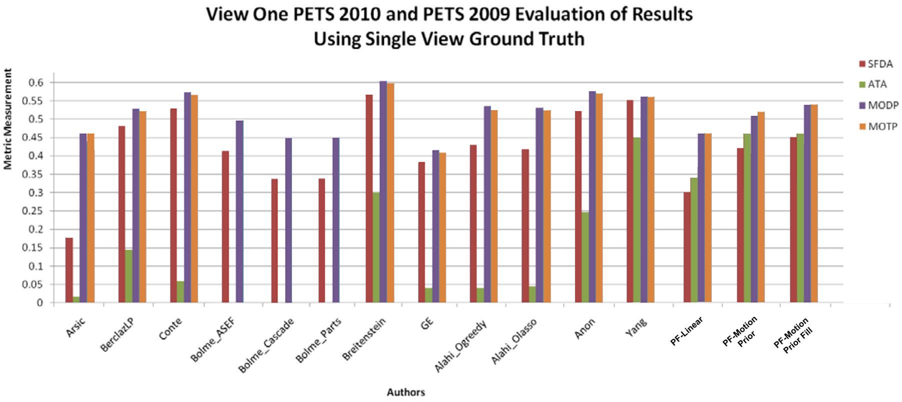


Fig. 4. Performance evaluation of tracking proposed by other authors and our proposal in set S2.L1 (view 1) of PETS 2009 dataset, for the four quality indicators. The last three results use our tracking system with linear motion model, motion prior model and a motion prior model with the strategy of filling zones with incomplete information. Other authors results have been reported in [12].

these prior distributions from optical flow computed on video-sequences grabbed by the same camera from which the camera is done. As in general few data are available for evaluating these distributions, we have proposed a regularization scheme for estimating them and filtering out artifacts. Then we have shown that the use of these priors in the probabilistic motion model of the tracker particle filter allows improvements in the tracker performance, in particular in terms of trajectory targets lengths.

Among our ongoing and future works, we aim at learning higher level information about not only local motion in the scene, but also on long-term information about trajectory, i.e. goals, interactions... based on low-level image information such as optical flow.

References

1. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'99) (1999)
2. Comaniciu, D., Meer, P.: Mean shift analysis and applications. Computer Vision, 1999. The Proc. of the Seventh IEEE International Conf. on (1999) 1197–1203
3. Isard, M., Blake, A.: Condensation-conditional density propagation for visual tracking. International Journal of Computer Vision **29** (1998) 5–28
4. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: European Conference on Computer Vision. Volume 1. (2002) 661–675
5. Nummiaro, K., Koller-Meier, E., Svoboda, T., Roth, D., Van Gool, L.: Color-based object tracking in multi-cameras environment. In: 25th Pattern Recognition Symposium, DAGM. (2003)

6. Doucet, A., de Freitas, N., Gordon, N.: sequnetial Monte Carlo methods in practice. Springer-Verlag, New York (2001)
7. Okuma, K., Taleghani, A., Freitas, N., Little, J., Lowe, D.: A boosted particle filter: multitarget detection and tracking. In: European Conference on Computer Vision. (2004) 28–39
8. Perez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. Proc. of the IEEE **92** (2004) 495– 513
9. Farneback, G.: Fast and accurate motion estimation using orientation tensors and parametric motion models. In: Proc. of Int. Conf. on Pattern Recognition. (2000) 135–139
10. Bouguet, J.: Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. In: USENIX Technical Conference. (1999)
11. Shi, J., Tomasi, C.: Good features to track. In: Int. Conf. on Computer Vision and Pattern Recognition (CVPR'94). (1994) 593 – 600
12. 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
13. 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)
14. Hu, W., et al.: Principal axis-based correspondence between multiple cameras for people tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence **28** (2006) 663–671
15. Doucet, A., De Freitas, N., Gordon, N., eds.: Sequential Monte Carlo methods in practice. Springer (2001)