A memory-based particle filter for visual tracking through occlusions

Antonio S. Montemayor, Juan José Pantrigo, Javier Hernández

Departamento de Ciencias de la Computación
Universidad Rey Juan Carlos, C/Tulipán, s/n,
28933 Móstoles, Madrid, Spain
antonio.sanz@urjc.es, juanjose.pantrigo@urjc.es,
j.hernandezsa@alumnos.urjc.es

Abstract. Visual detection and target tracking are interdisciplinary tasks oriented to estimate the state of moving objects in an image sequence. There are different techniques focused on this problem. It is worth highlighting particle filters and Kalman filters as two of the most important tracking algorithms in the literature. In this paper, we presented a visual tracking algorithm which combines the particle filter framework with memory strategies to handle occlusions, called as memory-based particle filter (MbPF). The proposed algorithm follows the classical particle filter stages when a confidence measurement can be obtained from the system. Otherwise, a memory-based module try to estimate the hidden target state and to predict its future states using the process history. Experimental results showed that the performance of the MbPF is better than a standard particle filter when dealing with occlusion situations.

1 Introduction

Visual tracking consists of locating or determining the configuration of a known (moving, deforming) object at each frame of a video sequence [5]. This is a relevant problem in Computer Vision and it has been focused using different methodologies. One of the most popular approaches in recent years is the particle filter (PF) proposed in [4].

Particle filter has demonstrated as an efficient method in visual tracking. Many works in the literature have proposed extensions to the original framework to deal with some specific difficult problems, such as tracking in cluttered environments, multi-dimensional or multiple object tracking, tracking through occlusions, etc. In this work we are specifically interested in tracking through occlusions. Particle filter algorithms for visual tracking need from a confidence measurement which characterizes the image region associated to the target. If the target is occluded, there are no target measurement in the image, or it is very poor, the standard particle filter algorithm will fail. A typical strategy consists of restart the tracking algorithm. Nevertheless, this is not always the best solution and there are many works in the literature which attemp to deal with occlusions without restarting the system. Bagdanov et al. [2] presents a continuously adaptive approach to estimating uncertainty in the particle filter to deal with the problem of undesired uncertainty amplifications in the model update which could lead to erroneous behavior of the tracker. Results obtained on a set of image sequences show that the performance of the particle filter is significantly improved through adaptive parameter estimation, particularly in cases of occlusions and nonlinear target motion. Wang et al. [7] proposes a multi-regions based particle filters for dealing with occlusion problems. The algorithm uses several nearly independent particle filters (NIPF) to track each region which will be influenced by the proximity and/or behavior of other regions. The authors claim that the proposed algorithm is more effective in solving long-time partial or total occlusion problem than other proposal in the literature. Ryu and Huber [6] presents an extension to the Particle Filter algorithm for tracking multiple objects. This approach instantiates separate particle filters for each object and explicitly handles partial and complete occlusion, as well as the instantiation and removal of filters in case new objects enter the scene or previously tracked objects are removed. The experiments demonstrate that the proposed method effectively and precisely tracks multiple targets and can successfully instantiate and remove filters of objects that enter or leave the image area.

The aim of this work is to extend the particle filter framework to estimate the state of a target even when it is occluded. To this aim we propose a memory-based particle filter (MbPF). The proposed state memory is inspired by the human visual perceptive system as far as humans are predisposed to track having objects of interest while they are visible and to predict their trajectories from past observations when they are occluded. This algorithm follows the classical particle filter stages when a confidence measurement can be obtained from the system. Otherwise, a memory-based module tries to estimate the hidden target state and to predict the future states using historic estimates.

The rest of the paper is organized as follows. Section 2 describes the particle filter framework. Section 3 presents the memory-based particle filter. Section 4 are devoted to present the obtained experimental results and, finally, Section 5 illustrates the conclusions and future works.

2 Particle filters for visual tracking

Sequential Monte Carlo algorithms (also called Particle Filters) are a specific class of filters in which theoretical distributions in the state-space are approximated by simulated random measures (also called particles) [3]. The state-space model consists of two processes: (i) an observation process $p(Z_{1:t}|X_t)$ where X_t denotes the system state vector and Z_t is the observation vector at time t, and (ii) a transition process $p(X_t|X_{t-1})$. Assuming that observations $\{Z_0, Z_1, \ldots, Z_t\}$ are sequentially measured in time, the goal is the estimation of the new system state at each time step. In the framework of Sequential Bayesian Modeling, the posterior *pdf* is estimated in two stages:

(a) Evaluation: the posterior *pdf* $p(X_t|Z_{1:t})$ is computed using the observation vector $Z_{1:t}$:

$$p(X_t|Z_{1:t}) = \frac{p(Z_t|X_t)p(X_t|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})} \tag{1}$$

(b) Prediction: the posterior *pdf* $p(X_t|Z_{1:t-1})$ is propagated at time step t using the Chapman-Kolmogorov equation:

$$p(X_t|Z_{1:t-1}) = \int p(X_t|X_{t-1})p(X_{t-1}|Z_{1:t-1})dX_{t-1} \tag{2}$$

A predefined system model is used to obtain an updated particle set. The problem lies in a state modeling where the dynamics equation describes the evolution of the object and the measurement equation links the observation with the state vector. Depending on the concrete application some choices are considered.

The aim of the PF algorithm is the recursive estimation of the posterior *pdf* $p(X_t|Z_{1:t})$, that constitutes a complete solution to the sequential estimation problem. This *pdf* is represented by a set of weighted particles $\{(\mathbf{x}_t^0, \pi_t^0), \ldots, (\mathbf{x}_t^N, \pi_t^N)\}$, where the weights $\pi_t^i = p(Z_{1:t}|X_t = \mathbf{x}_t^i)$ are normalized. Each particle i stores a system state $\mathbf{x}_t^i$ at time t and a quality measure π_t^i called weight, proportional to the probability of the state $\mathbf{x}_t^i$.

The PF algorithm starts by initializing a population vector X_0 of N particles using a known *pdf*. The measurement vector Z_t at time step t is obtained from the system, and particle weights Π_t are computed using a fitness function. The weights are normalized and a new particle set X_t^* is selected. Taking into account that particles with larger weight values can be chosen several times, a diffusion stage is applied to avoid the loss of diversity in X_t^*. Finally, particle set at time step $t+1$, X_{t+1}, is predicted using the motion model. The pseudocode of a general PF is detailed in [1].

In short, Particle Filters are algorithms that handle the evolution of particles. Particles in PF are driven by the state model and are multiplied or eliminated according to their fitness values (weights) as determined by the *pdf* [3]. In visual tracking problems, this *pdf* represents the probability that the object is in a determined position and/or orientation in the frame.

3 The memory-based particle filter

Figure 1 shows the memory-based particle filter (MbPF) algorithm scheme. The proposed algorithm follows a Particle Filter scheme (see section 2 for a detailed explanation) except when the tracked target is occluded. The estimation that an occlusion takes place will be described next. In this case, an alternative strategy based on the history of the process is used. Past estimations by the current time t_c are stored in a set $\{\hat{\mathbf{s}}_t,\ t_c - T_M \leq t \leq t_c\}$ where $\hat{\mathbf{s}}_t$ is the estimated target state at time t and T_M is the selectable number of frames of the length of the memory. In this work we consider a target state given by its 2D position ($\hat{\mathbf{s}}_t = [\hat{x}_t, \hat{y}_t]$). In

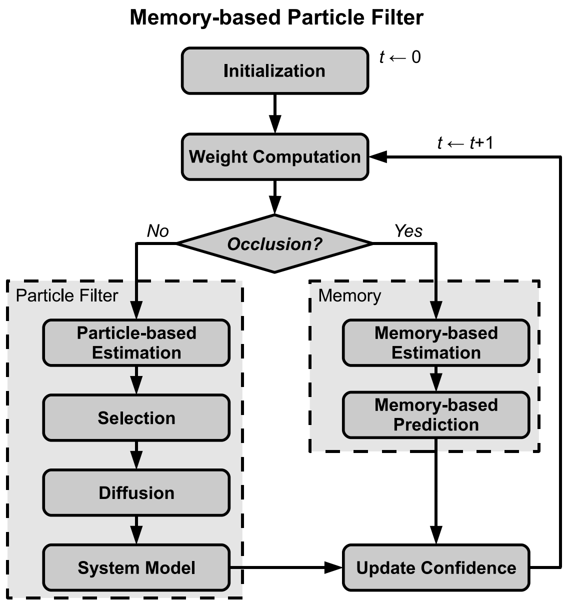

Fig. 1. Algorithm overview.

the same way, the state $\mathbf{s}^p$ of each particle p in the particle set is given by the dupla $\mathbf{s}^p = [x^p, y^p]$ and an associate weight π^p. The rest of this section analyzes the succesive involved modules in detail.

3.1 Initialization

The aim of the stage is to provide initial values to the particles state. This initial stage is performed only once, at time $t = 0$. As a result, each particle p in the particle set randomly initializes its state $[x^p, y^p]$ over the whole image as:

$$\begin{cases} x_0^p = R([0, W]) \\ y_0^p = R([0, H]) \end{cases} \tag{3}$$

where R is a random uniform variable in a given range (in this case, $[0, W]$ or $[0, H]$), W and H are the image length and width, respectively, and $p \in [1, N]$, where N is the number of particles in the particle set.

3.2 Weight computation

This subtask receives a segmented image I_M^t from the system at time t. I_M^t is a binary image in which white pixels correspond to target and black pixels correspond to other image regions. The weight π_t^p assigned to each state $\mathbf{s}_t^p$ of the particle p at time t is computed summing up the number of white pixels that belongs to a predefined object bounding box in the measurement image I_M^t:

$$\pi_t^p = \sum_{w=x_t^p-(Lx/2)}^{x_t^p+(Lx/2)} \left(\sum_{h=y^s-(Ly/2)}^{y^s+(Ly/2)} I_M^t(w,h) \right) \tag{4}$$

where Lx and Ly are the size of the predefined bounding box. The higher the number of white pixels contained in the object bounding box, the higher the likeliness of the particle is.

3.3 Occlusion condition

We consider a target is occluded when there are no particles in the particle set with a weight higher than a given threshold. In other words:

$$Occlusion? = (\pi^p \leq th_o,\ \forall p \in [1,N]) \tag{5}$$

where N is the number of particles in the particle set and th_o is a predefined threshold.

3.4 Particle-based estimation

The particle-based estimation is computed as the state of the particle in the particle set with maximum weight. In mathematical terms, the estimation at time t is given by:

$$\hat{\mathbf{s}}_t = argmax_{\pi^p}(\{\mathbf{s}^p,\ \forall p \in [1,N]\}) \tag{6}$$

where N is the number of particles in the particle set.

3.5 Selection

Particle set for the next time step $t+1$ is made up of particles selected from the particle set at time t. Particles are selected with probabilities according to their weights.

3.6 Diffusion

The previous selection stage may select the same particle several times. The PF diffusion method is used to keep the needed diversity in the particle set once the selection stage was performed. This diffusion basically consists of a random perturbation of the state of every particle:

$$\begin{cases} x'^p = x^p + R([-r,r]) \\ y'^p = y^p + R([-r,r]) \end{cases} \tag{7}$$

where x, y and x', y' denote the spatial variables before and after the perturbation, respectively, and $R(-r,r)$ is a random uniform variable in a predefined range $[-r,r]$.

3.7 System Model

The system model describes the temporal update rule for the system state [8]. The tracked object state consists of a given number of spatial coordinates and their corresponding velocities. In mathematical terms, the update rule can be expressed as follows:

$$\begin{cases} x^p_{t+\delta t} = x^p_t + \dot{x}^p_t \delta t + R([-r, r]) \\ y^p_{t+\delta t} = y^p_t + \dot{y}^p_t \delta t + R([-r, r]) \\ \dot{x}^p_{t+\delta t} = \dot{x}^p_t + R([-vr, vr]) \\ \dot{y}^p_{t+\delta t} = \dot{y}^p_t + R([-vr, vr]) \end{cases} \tag{8}$$

where x, y denote the spatial variables, $\dot{x}, \dot{y}$ are the first derivatives of x, y with respect to t, δt is the time step and R is a random uniform variable in a predefined range, which allow changes in the object state in the ranges $[-r, r]$ for the position and $[-vr, vr]$ for the velocity. The values of r and vr depend on the expected changes in the position and velocity of the tracked object.

3.8 Memory-based estimation

The memory-based estimation is performed when no confidence measurement of the system state is available. This situation is typically arises when an occlusion occurs. Then, the proposed algorithm trusts the system state history more than the current measurement. Let $S_M = \{\hat{\mathbf{s}}_t, \quad t \in [t_{c-1} - T_M, t_{c-1}]\}$ the set of estimates stored in the history of the process up to the current time step t_c. We compute the estimate at current time t_c by means of a well-known least squares method. To achieve this goal, we first compute the least square line for each variable of the S_M set:

$$\begin{cases} x(t) = a_x t + b_x \\ y(t) = a_y t + b_y \end{cases} \tag{9}$$

where a_x, b_x, a_y, b_y are the coefficients obtained by means of the least squares method. Finally, we obtain the system estimate $\hat{\mathbf{s}}_{t_c} = [\hat{x}_{t_c}, \hat{y}_{t_c}]$ at time t_c, by replacing $t = t_c$ in the former expressions, obtaining:

$$\begin{cases} \hat{x}_{t_c} = x(t_c) = a_x t_c + b_x \\ \hat{y}_{t_c} = y(t_c) = a_y t_c + b_y \end{cases} \tag{10}$$

3.9 Memory-based prediction

The memory-based prediction consists of the computation of a bounding box containing the object in the next time step. The size of this bounding box depends on a confidence estimation. This confidence decreases when there is no measurement available. It gets even lower as the number of frames without observation increases once we lost the target or it was occluded. This bounding box will be greater at each time step while the confidence decreases.

The predicted bounding box is defined by a position of its geometrical center $[x_t^{BB}, y_t^{BB}]$ and a size $[Lx_t^{BB}, Ly_t^{BB}]$. The position $[x_t^{BB}, y_t^{BB}]$ is predicted following the same philosophy of the previous memory-based estimation stage. The size $[Lx_t^{BB}, Ly_t^{BB}]$ is updated from an initial predefined size $[Lx_0^{BB}, Ly_0^{BB}]$, using a confidence measurement C_t at time t, as follows:

$$\begin{cases} Lx_t^{BB} = Lx_0^{BB} + C_t \times \delta Lx \\ Ly_t^{BB} = Ly_0^{BB} + C_t \times \delta Ly \end{cases} \tag{11}$$

where $[\delta Lx, \delta Ly]$ are predefined increments for the bounding box width and height, respectively. Next section explains how to update the confidence measurement C_t.

3.10 Confidence measurement update rule

The confidence measurement is updated at every time step, according to the following rule:

$$C_{t+1} = \begin{cases} C_t + \delta C, & \text{if hidden target} \\ 0, & \text{if visible target} \end{cases}$$

where δC is a predefined parameter which models the loss of confidence when no object evidence is achieved by the measurement model in the current frame.

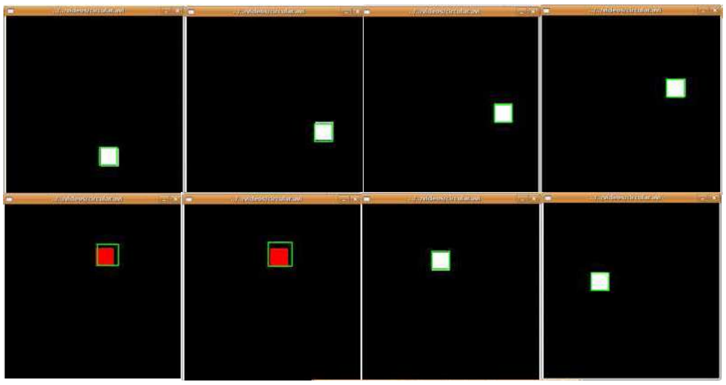

Fig. 2. Some non-consecutive frames extracted from the circular sequence.

4 Experimental results

This section is devoted to present and discuss the obtained experimental results. The proposed algorithm memory-based particle filter (MbPF) has been tested

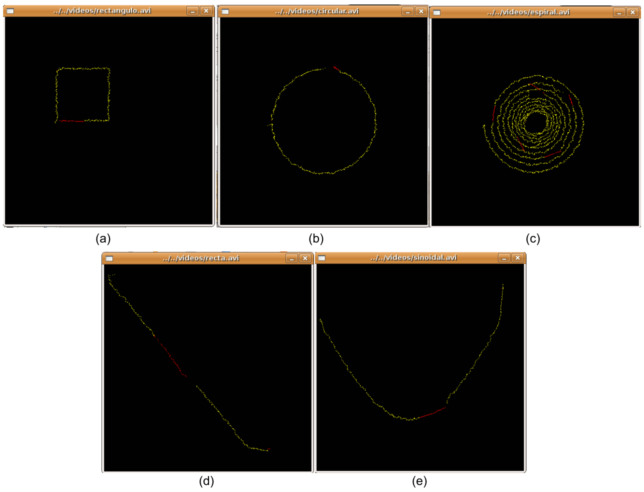

Fig. 3. Obtained results in different trajectories: (a) square, (b) circle, (c) spiral, (d) line and (e) sinusoidal. Yellow and red points represent visible and hidden target estimations, respectively.

for the single object visual tracking in synthetic sequences. In these sequences, the object appears as a white squared region when it is visible and as a square red region when it is hidden (see Figure 2). The red regions in the image are not detected as object by the measurement model although it is shown for easy verification. Therefore, this represents an occlusion situation for the tracking algorithm. The tracked target follows a predefined trajectory. We have tested five different trajectories: circular, rectangular, sinusoidal, spiral and linear. Figure 3 shows the trajectory achieved and predicted by the MbPF. The estimations performed by the MbPF when the target is visible are represented by yellow points, while the estimations performed through occlusions are represented by red points.

In order to have a comparison baseline, a standard particle filter (PF) has been tested in the same experimental conditions. We have measured the accuracy of PF and MbPF in the five image sequences presented above. The accuracy measurement is computed as the number of target pixels that belongs to the estimate bounding box. In mathematical terms, the accuracy $A(\hat{\mathbf{s}}_t)$ of the estimate $\hat{\mathbf{s}}_t)$ at time t is computed as follows:

$$A(\hat{\mathbf{s}}_t) = \frac{\sum_{w=\hat{x}_t-(Lx/2)}^{\hat{x}_t+(Lx/2)} \left(\sum_{h=\hat{y}_t-(Ly/2)}^{\hat{y}_t+(Ly/2)} I_M^t(w,h) \right)}{Lx \times Ly} \tag{12}$$

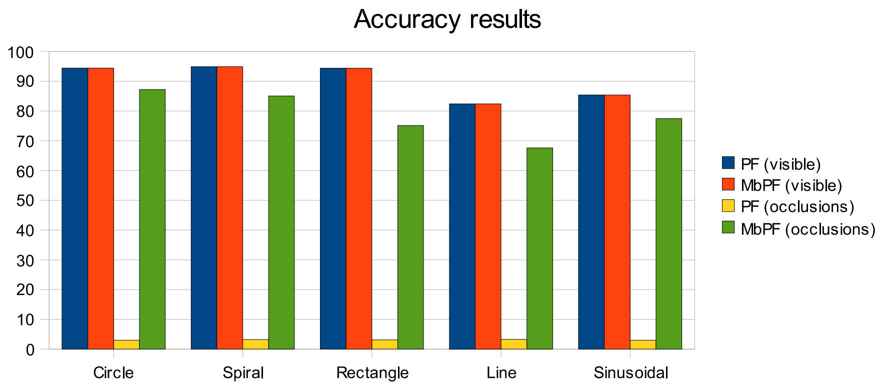

Fig. 4. Accuracy results in different trajectories obtained by PF and MbPF, when the target is visible and when it is hidden.

where I_M^t measurement image and parameters $[Lx, Ly]$ are the size of the predefined bounding box.

Figure 4 shows the average accuracy results obtained by PF and MbPF when the target is visible and hidden in the considered sequences. As it can be seen in the figure, PF and MbPF obtain the same results when the object is visible. However, the accuracy of the MbPF is much better than PF through occlusions. As a result, the average accuracy results for the whole sequences are also in favour of our proposal, as it is depicted in Figure 5.

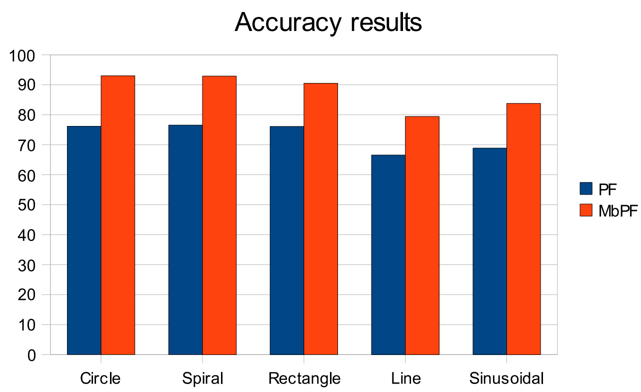

Fig. 5. Accuracy results in different trajectories obtained by PF and MbPF.

5 Conclusion and future work

In this paper, we presented a visual tracking algorithm which combines the particle filter framework with memory strategies to handle occlusions, called as memory-based particle filter (MbPF). The proposed algorithm follows the classical particle filter stages when a confidence measurement can be obtained from the system. Otherwise, a memory-based module tries to estimate the hidden target state and to predict the future states using the process history. The performance of the MbPF has been compared with a standard particle filter (PF) in image sequences in which a target is tracked through occlusions. In spite of its simplicity, MbPF achieves very promising results on the tested image sequences, demonstrating better performance than a PF in all the considered experiments. These experimental results allow us to be optimistic about the MbPF application in real environments.

In a future work, we will extend the proposed algorithm to handle multi-dimensional problems. We are particularly interested in the multiple object tracking problem with occulsions. Finally, we will apply learning methods to improve the performance of the memory-based stages.

Acknowledgements

This research has been partially supported by the Spanish projects TIN2008-06890-C02-02 and URJC-CM-2008-CET-3625.

References

1. Arulampalam S.M., Maskell S., Gordon N., and Clapp T. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking". IEEE Trans. on Signal Processing, 50:2, 174-178 (2002)
2. Bagdanov, A.D., Del Bimbo, A., Dini, F., Nunziati, W. "Adaptive uncertainty estimation for particle filter-based trackers". 14th International Conference on Image Analysis and Processing (ICIAP 2007) 331-336, 2007.
3. Carpenter, J., Clifford, P., Fearnhead, P. "Building robust simulation based filters for evolving data sets". Tech. Rep., Dept. Statist., Univ. Oxford, Oxford, U.K (1999)
4. Gordon, N.J., Salmond, D.J. and Smith, A.F.M. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". IEE Proceedings F Radar & Signal Processing 140:2, 107-113, 1993.
5. MacCormick, J. "Stochastic Algorithm for visual tracking". Springer, 2002.
6. Ryu, H., Huber, M. "A Particle Filter Approach for Multi-Target Tracking". Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007.
7. Wang, Y., Zhao. W., Liu, J., Tang, X., Liu, P. "A novel particle filter based people tracking method through occlusion". Proceedings of the 11th Joint Conference on Information Sciences, 2008.
8. Zotkin, D., Duraiswami, R. and Davis, L. "Joint Audio-Visual Tracking Using Particle Filters". EURASIP Journal on Applied Signal Processing. 11, 1154-1164, 2002.