Probabilistic Image-Based Tracking: Improving Particle Filtering

Daniel Rowe, Ignasi Rius, Jordi Gonzàlez, Xavier Roca, and Juan J. Villanueva

Computer Vision Centre/Department of Computing Science
Universitat Autònoma de Barcelona. 08193 Bellaterra, Barcelona, Spain
drowe@cvc.uab.es

Abstract. *Condensation* is a widely-used tracking algorithm based on particle filters. Although some results have been achieved, it has several unpleasant behaviours. In this paper, we highlight these misbehaviours and propose two improvements. A new weight assignment, which avoids sample impoverishment, is presented. Subsequently, the prediction process is enhanced. The proposal has been successfully tested using synthetic data, which reproduces some of the main difficulties a tracker must deal with.

1 Introduction

The increasing interest in visual tracking is motivated by a huge number of promising applications that can now be tackled in real time thanks to recent technological advances. These applications include performance analysis, surveillance, video-indexing, smart interfaces, teleconferencing and video compression.

However, tracking agents can be extremely complex and time-consuming. To start with, strong requirements are mandatory. Real-time processing, extreme robust performances or high accuracy may be critical. On the other hand, difficulties common to all vision areas could cause system failures, specially in open environments. Hence, several of the following premises are often assumed: we can consider outdoors or indoors scenes, static or in-motion background, illumination changes, shadows, presence of clutter or a-priori known objects. Some foreground assumptions are also taken into account concerning whether a single or multiple agents should be expected; agents entries and exits from the scene; smooth, restricted or already-known dynamics; occlusions; carried objects; or appearance changes.

This paper focuses on solving some tracking problems related to the difficulties described above, such as multiple-agent tracking with unknown dynamics in presence of background clutter and strong noise. Specifically, we present some improvements to a well-known tracking algorithm, *Condensation* [3].

The remainder of this paper is organized as follows. Section 2 covers the probabilistic framework, revises *Condensation*, exposing its misbehaviours, and reviews a *Condensation*-based algorithm called *iTrack* [7]. Section 3 proposes several improvements on *Condensation*/*iTrack*. Section 4 shows experimental results with synthetic data and section 5 concludes this paper.

J.S. Marques et al. (Eds.): IbPRIA 2005, LNCS 3522, pp. 85–92, 2005.

2 Image-Based Probabilistic Tracking

The **environment** is composed of agents, static objects and background conditions. The **scene** is defined as the piece of environment which a visual sensor can capture. The aim of the tracking task is to estimate the scene **state** over time. In this context, the state will be the parameterised knowledge which will characterise the scene evolution. Due to practical and theoretical ignorance, we do not have access to the ground truth. A probabilistic framework is commonly used as a way to perform tracking [5]. Classical approaches, such as the **Kalman Filter**, rely on linearity and gaussianity assumptions about the involved distributions. More recent works make use of **Bayesian filters** combined with **Monte Carlo Simulation** methods in order to deal with nonlinear and non-Gaussian transition models [1, 2]. Subsequent developments have introduced a re-sampling phase in the sequential simulation-based Bayesian filter algorithms. Such methods were first introduced in computer vision in *Condensation* [3]. However, they have several important drawbacks as stated in [4]. A great number of improvements have been introduced in recent years [6, 7] but there is still much ground to cover before solving unconstrained tracking.

2.1 Bayesian Filtering

The computation of the belief state $\mathbf{S}_t$ given all evidence to date $\mathbf{e}_{1:t}$ is called **filtering.** The posterior pdf[1] can be calculated through **recursive estimation**:

$$P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) = \underbrace{\underbrace{P\left(\mathbf{e}_t \mid \mathbf{S}_t\right)}_{\text{likelihood}}}_{\text{updating}} \underbrace{\sum_{\mathbf{s}_{t-1}} \underbrace{P\left(\mathbf{S}_t \mid \mathbf{s}_{t-1}\right)}_{\text{transition mod.}} \underbrace{p\left(\mathbf{s}_{t-1} \mid \mathbf{e}_{1:t-1}\right).}_{\text{previous post.}}}_{\text{prediction}} \tag{1}$$

The pdf is projected forward according to the transition model, making a prediction, and it is updated in agreement with the likelihood function value based on the new evidence.

2.2 Condensation

Recursive estimation leads to expressions that are impossible to evaluate analytically unless strong assumptions are applied. *Condensation* addresses filtering when no assumption about linearity or gaussianity is made [3]. This problem is overcome by simulating N independent and identically-distributed samples from the posterior pdf, $\left\{\mathbf{s}_t^i; i = 1:N\right\}$. The temporal prior$\left\{\hat{\mathbf{s}}_t^i\right\}$ is obtained by applying the transition model to each sample. Weights π_t^i are assigned according to the likelihood function. Once all samples have been propagated and measured,

[1] Notation: bold case denotes vectors and matrices whereas non-bold case denotes scalars. Matrices are in uppercase. In a probabilistic context, uppercase denotes probability density functions (pdf) and random variables; lowercase denotes probabilities and variable instances. $\mathbf{X}_{a:b}$ denotes a variable set from time $t = a$ to $t = b$.

the set is re-sampled using normalized weights $\overline{\pi}_t^i$ as probabilities. This sample set represents the new posterior. Expectations can be approximated as:

$$\mathbb{E}_{P(\mathbf{S}_t|e_{1:t})}\left(\mathbf{S}_t\right) \simeq \sum_{i=1}^{N} \overline{\pi}_t^i \hat{\mathbf{s}}_t^i = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_t^i. \tag{2}$$

However, it has several unpleasant behaviours as stated in [4]. **Sampling impoverishment** is one of the main drawbacks of re-sampling algorithms. Samples are spread around several **modes** indicating hypotheses in the space state. Nevertheless, some of them are spurious. Similarly to genetic drift, there is a non-negligible probability of losing modes, a low probability of recovering them and the remaining modes could be all spurious. It can also be derived from this fact that different runs of the algorithm lead to different results. Therefore, computed expectations in different runs have high variance although computed expectations within the same algorithm run have low variance making the tracker look stable. On the other hand, *Condensation* has a tendency of clustering samples even when the likelihood function gives no information at all. In addition, the sample set size N is kept constant over time. Unfortunately, there is no information about how large N should be for a requested precision. Once N have been heuristically set, it may happen that at later times larger values of N may be required. Finally, *Condensation* was designed to keep multiple-hypothesis for a single agent.

2.3 iTrack

iTrack is a visual tracking algorithm based on *Condensation* [7], but both transition model and likelihood function are redefined. It also introduces some improvements in order to overcome some *Condensation* drawbacks and cope with multiple agents.

iTrack uses a first-order dynamic model in image coordinates to model the motion of the central point of a bounding box. The l-labeled agent's state is defined as $\mathbf{s}_t^l = (\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t, \mathbf{A}_t)^T$ where each element represents the position, speed, bounding-box size and pixel appearance, respectively. The label associates one specific appearance model to the corresponding samples, allowing multiple-agent tracking. However, multiple-agent tracking causes several problems including that the agent with higher likelihood monopolizes the sample set. Denoting as N_j the number of samples belonging to the l-labeled agent, *iTrack* proposed the following normalization to avoid this issue:

$$\overline{\pi}_t^{i,l} = \frac{\pi_t^{i,l}}{\sum_{i\,=\,1}^{N} \pi_t^{i,j}} \frac{N_j}{N}, \quad \text{where } j = l. \tag{3}$$

An initial pdf, provided by a segmentation method, is needed to start the recursive estimation. *iTrack* also uses this pdf to reinitialize the algorithm allowing multiple-agent tracking and error recovery. Thus, some samples are generated according to the prior instead of being propagated.

3 Improving Condensation/iTrack

3.1 Improvement 1. Sampling Impoverishment

Whether data association is feasible, using the prior density to generate new samples reduces the risk of sampling impoverishment. However, it is not completely avoided, since it depends on the probability of generating new samples, on whether these new samples represent the extincting mode, and on whether they can be associated to it. This problem is increased in a multiple-agent tracking scenario. Without considering new sample generation, losing an agent track is only a matter of time, according to the sample set size. In this case, those agents whose samples exhibit lower likelihood have higher probability of being lost, since the probability of propagating one mode is proportional to the cumulative weights of the samples that constitute it. Two kind of modes can be distinguished. In the first place, samples with different labels belong to different modes. Thus, several agents can be tracked simultaneously. Secondly, samples with the same label could be spread around different modes. This fact allows us to keep several hypotheses. Hopefully, one of them represents the true agent state and the others are due to background clutter.

In order to avoid single agent modes absorbing other agent samples, *genetic drift* must be prevented. This fact happens due to the lack of *genetic memory*: we propose to include a memory term which takes into account the number of agents being tracked. Hence, weights are normalized according to:

$$\overline{\pi}_t^{i,l} = \frac{\pi_t^{i,l}}{\sum_{i=1}^{N} \pi_t^{i,j}} \frac{1}{N_a}, \quad \text{where } j = l, \tag{4}$$

where N_a is the number of agents being tracked. It does not assign a fixed number of samples to each agent but ensures that each agent will have the same probability of being propagated. Furthermore, it can be combined with new sample generation, thereby improving the general performance. On the other hand, modes due to clutter are pruned because of differences in their dynamics. It is unlikely that any sample tracks local clutter since it implies highly abrupt changes in the dynamics. Non-losing the true mode depends on how accurate the dynamic model is, and how the different hypotheses are generated.

3.2 Improvement 2. Agent Dynamics

iTrack makes predictions according to the following expressions:

$$\hat{\mathbf{x}}_t^i = \mathbf{x}_{t-1}^i + \mathbf{u}_{t-1}^i \Delta t + \xi_x^i, \qquad \hat{\mathbf{u}}_t^i = \mathbf{u}_{t-1}^i + \xi_u^i\,. \tag{5}$$

The random terms ξ_x^i, ξ_u^i provide the system with a diversity of hypothesis. Samples with high likelihood are supposed to be propagated. Sample likelihoods depend on samples position but they do not depend on their speed. Thus, propagated samples could have an accurate position, but their speed values become

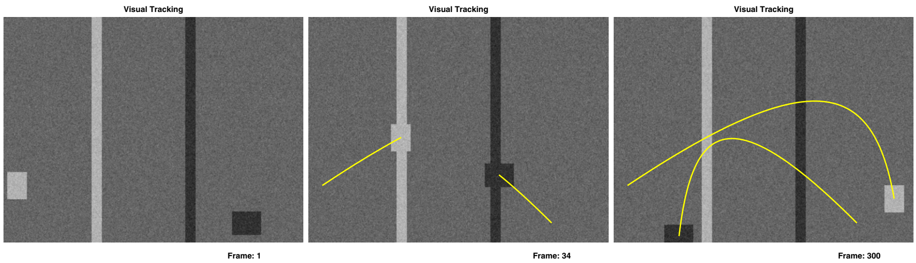


Fig. 1. Ground Truth

completely different from the agent's one in a few frames. Agents could be tracked since we are in a multiple-hypothesis scenario, but an important proportion of samples are wasted. The j-agent state is estimated according to:

$$\hat{\mathbf{s}}_t^j = \frac{1}{N_j} \sum\nolimits_{i=1}^{N} \mathbf{s}_t^{i,j}. \tag{6}$$

Our approach proposes to feed-back the estimated agent speed at time $t-1$, denoted as $\hat{\mathbf{u}}_{t-1}^j$, into the prediction:

$$\hat{\mathbf{u}}_t^{i,j} = \hat{\mathbf{u}}_{t-1}^j + \xi_u^i. \tag{7}$$

However, there is still a weak relation between the agent and the estimated speeds: they are chosen only due to the sample weights, which do not depend on the current speed. We propose to enhance the estimation by considering not only the estimated speed from the selected samples but also by calculating the instant speed according to the history of positions. The following expressions update the agent position and speed recursively considering this fact:

$$\begin{aligned} \hat{\mathbf{x}}_t^j &= \hat{\mathbf{x}}_{t-1}^j (1 - \alpha_p) + \left(\frac{1}{N_j} \sum\nolimits_{i=1}^{N} \mathbf{x}_t^{i,j} \right) \alpha_p, \\ \hat{\mathbf{u}}_t^j &= \hat{\mathbf{u}}_{t-1}^j (1 - \alpha_s) + \left(\hat{\mathbf{x}}_t^j - \hat{\mathbf{x}}_{t-1}^j \right) \alpha_s, \end{aligned} \tag{8}$$

where α_p, α_s denote the adaptation rates. The estimated speed is then fed-back when predicting the following sample state.

4 Experimental Results

In order to evaluate the algorithm performance, a two-moving-agent synthetic experiment has been designed. The aim is to cover several difficulties a tracker can run into, see Fig. 1. The background pixel intensity values are set randomly following a normal distribution. Both agents' pixel intensity values also have a normal distribution around different means. Two vertical strips are drawn in the background, simulating heavy clutter. Their distributions are identical to both agent's ones, thereby mimicking them. Strong acquisition-device noise, modeled

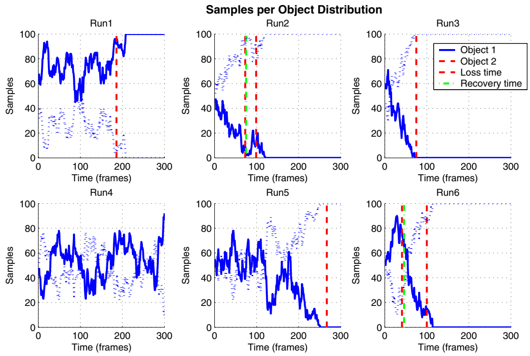


Fig. 2. *Condensation/iTrack* performance

Table 1. Performance of improvement 1

Mean normalized error		
	Agent 1	Agent 2
Run 1	0.1163	0.1309
Run 2	3.8864	0.1182
Run 3	0.1222	0.1226
Run 4	0.0980	0.1038
Run 5	0.1612	0.1131
Run 6	0.1101	2.4679

Table 2. Performance of improvement 1, 2

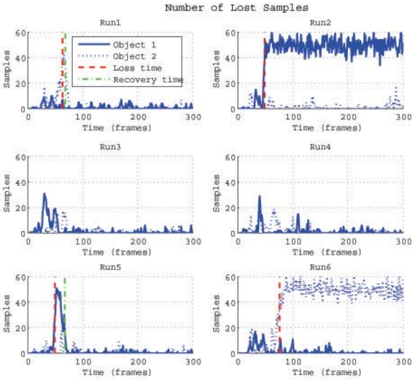
Mean normalized error		
	Agent 1	Agent 2
Run 1	0.0715	0.0716
Run 2	0.0849	0.1163
Run 3	0.0987	0.1289
Run 4	0.0645	0.0595
Run 5	0.0679	0.1173
Run 6	0.1233	0.0840

as White Additive Gaussian Noise, is simulated[2]. A highly non-linear dynamic is simulated: both agents move as projectiles which are shot into an environment with gravity and air friction. Tracking is performed over $T = 300$ frames using $N = 100$ samples. We present results of six random runs for each of the three approaches considered, namely, *iTrack* and both presented improvements. New sample generation is not used in order to evaluate only the tracking performance.

In 5 out of the 6 *iTrack* runs, an agent is lost due to the lack of samples, see Fig. 2. In the remaining one, at time $t = 300$ an agent got 92% of the samples. An agent is considered lost when the normalized Euclidean distance, according to the agent size, between the agent and the estimation position is higher than a threshold set at 0.5. On the other hand, after the proposed weight normalization, the mean number of samples per agent fluctuates between 49.5 % and 50.5%.

Table 1 shows the mean normalized error, according to the agent size, in the estimation of the agent position before applying the new dynamics updating whereas Table. 2 shows the same results after applying it. A significant error reduction can be appreciated. Figs. 3, 4 compare the number of samples per agent that had lost the agent. After considering this improvement, a significant sample loss reduction is observed. Furthermore, none of the agents is ever lost.

[2] The standard deviation is set at 0.03 which implies nearly a ten per cent deviation.



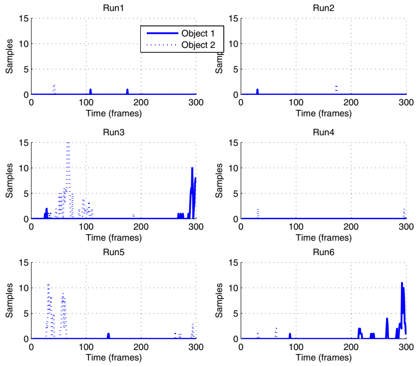
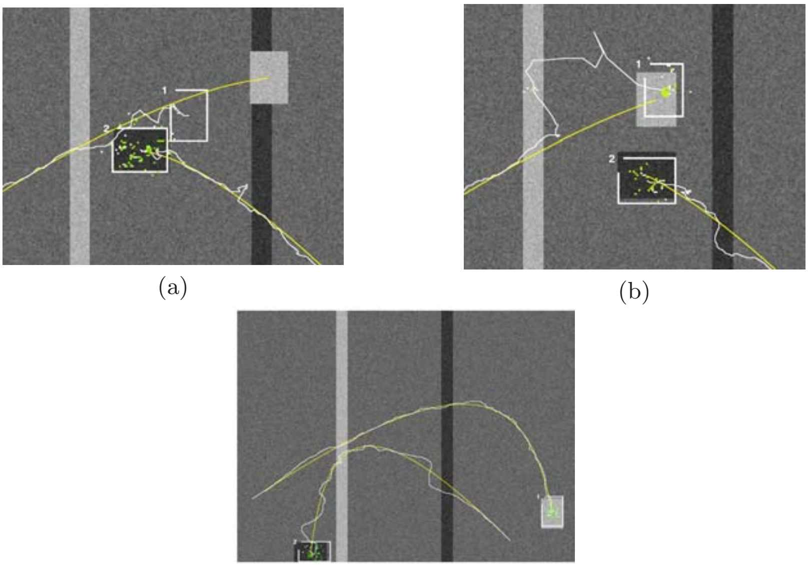


Fig. 3. Performance of improvement 1 **Fig. 4.** Performance of improvements 1, 2 (Notice that axes scale are reduced in 75%)



(a) (b) (c)

Fig. 5. Behaviour of the three studied trackers

The trackers behaviour can be seen in Fig. 5: Fig. 5.(a), corresponding to *iTrack*, shows how one of the agents absorbs all the samples. Fig. 5.(b), after applying the normalization improvement, shows agent recovery since the tracker have preserved enough samples to cope with multiple hypotheses. Thus, both modes, the agent and the clutter, are tracked until the clutter one disappears. Fig. 5.(c) shows the tracker performance once both improvements are considered.

5 Conclusions

In this paper, we have extended *Condensation* in order to enhance multiple-agent tracking. A new approach is taken to deal with one of *Condensation* great misbehaviours, the sampling impoverishment. This problem becomes critical in a multiple-tracking scenario. The new sample-weight normalization prevents from loosing any of the targets due to the lack of samples. The dynamics updating is modified by feed-backing the estimated speed into the prediction stage. The agent speed is estimated combining two sources of knowledge: the fittest sample speed and the position historic. Thanks to both improvements, the tracker copes successfully with multiple-agent tracking. These agents have a highly non-linear dynamics which is successfully tracked using a constant-speed approach. Moreover, it also deals with complex clutter, which mimics the agent appearances, and strong noise. Improvements shown in these synthetic experiments are currently being applied in real applications relative to traffic surveillance. Encouraging results are being achieved.

Acknowledgements

This work has been partially supported by the Spanish CICYT TIC 2003-08865.

References

1. Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Transactions on Signal Processing 50 (2): 174 - 188, 2002.
2. Doucet, A. On Sequential Simulation-Based Methods for Bayesian Filtering. CUED/F-INFENG/TR 310. University of Cambridge, 1998.
3. Isard, M. and Blake, A. Condensation - Conditional Density Propagation for Visual Tracking. International Journal of Computer Vision 29 (1): 2 - 18, 1998.
4. King, O and Forsyth, D. A. How Does Condensation Behave with a Finite Number of Samples? ECCV proceedings (1): 695 - 709, 2000.
5. Russell, R. and Norvig, P. Artificial Intelligence, a Modern Approach. Chapters 13-15. Prentice Hall, 2003.
6. Merwe, R. van der, Doucet, A., de Freitas, N. and Wan, E. The Unscented Particle Filter. CUED/F-INFENG/TR 380. University of Cambridge, 2000.
7. Varona, X., Gonzàlez, J., Roca, X. and Villanueva, J. J. *iTrack*: Image-based Probabilistic Tracking of People. ICPR (3): 7122 - 7125, 2000.