Tracking Multiple Objects with Particle Filtering

Carine Hue, Jean-Pierre Le Cadre, Patrick Perez

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Tracking Multiple Objects with Particle Filtering

Carine Hue , Jean-Pierre Le Cadre , Patrick Pérez

N˚4033

Octobre 2000

THÈME 3

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4033--FR+ENG

Tracking Multiple Objects with Particle Filtering

Carine Hue *, Jean-Pierre Le Cadre † , Patrick Pérez ‡

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Vista

Rapport de recherche n° 4033 — Octobre 2000 — 28 pages

Abstract: We address the problem of multitarget tracking encountered in many situations in signal or image processing. We consider stochastic dynamic systems detected by observation processes. The difficulty lies on the fact that the estimation of the states needs to know the assignation of the observations to the multiple targets. We propose an extension of the classical particle filter where the stochastic vector of assignation is estimated by a Gibbs sampler. This algorithm is used to estimate the trajectories of multiple targets from their noisy bearings, thus showing its ability to solve the data association problem. Moreover this algorithm can easily be extended to multireceiver observations where the receivers can produce measurements of various nature with different periodicities.

Key-words: Multitarget tracking, particle filter, Bayesian estimation, Gibbs sampler, multireceiver, bearings-only tracking.

(Résumé : tsvp)

* IRISA - Université de Rennes 1, Carine.Hue@irisa.fr
† IRISA - CNRS, lecadre@irisa.fr
‡ Microsoft Research Center, Cambridge, pperez@microsoft.com

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Suivi d'objets multiples par filtrage particulaire

Résumé : Nous traitons dans ce rapport le problème du suivi de plusieurs objets souvent rencontré en traitement du signal et en imagerie. Nous considérons des systèmes dynamiques détectés par des processus d'observations. La principale difficulté repose sur le fait que l'estimation des états nécessite de connaître les associations des observations aux objets. Nous proposons ici une extension du filtre particulaire classique dans lequel le vecteur de probabilité des associations est estimé par un échantilloneur de Gibbs. L'algorithme est utilisé pour estimer les trajectoires de plusieurs objets uniquement grâce à leurs mesures d'angles et prouve sa capacité à résoudre le problème d'association. De plus, l'algorithme peut être étendu au cas d'observations issus de capteurs différents pouvant produire leurs mesures avec des périodicités semblables ou différentes.

Mots-clé : Suivi multi-objets, filtrage particulaire, estimation Bayésienne, échantillonneur de Gibbs, multirécepteur, pistage par mesure d'angles.

1 Introduction

Multitarget tracking (MTT) deals with state estimation of an unknown number of (moving) targets. Available measurements may have originated from the targets if they are detected or of a special model called "clutter". Clutter is generally considered as a model describing false alarms. Its (spatio-temporal) statistical properties are quite different from target ones which makes possible the separation of target tracks on the one hand and clutter model on the second. To perform multitarget tracking the observer has at its disposal a huge amount of data, possibly collected on multiple receivers. Elementary measurements are receiver outputs, e.g., bearings, ranges, time-delays, Dopplers, etc.
But the main difficulty comes from the assignment of a given measurement to a target model. For critical situations, these assignments are unknown, as are the true target models. This is a neat departure from classical estimation problems. Thus, two distinct problems have to be solved *jointly* : data association and estimation.
The simpler approach is probably the Nearest Neighbor approach. Using only the observation the closest to the predicted state, the algorithm is not robust enough in many situations. More elaborated, the Joint PDAF ([6]), extension of the Probabilistic Data Association Filter for multiple objects ([1]), estimates the states by a sum over all the associations weighted by the probabilities that a measurement is associated to a target. Only the associations at one time period are there considered. The Multiple Hypothesis Tracker recursively builds the association hypothesis and assumes that each measurement can be issued from a new source. Because these two algorithms assume that one target produces only one measure, the set of associations to be enumerated grows exponentially and becomes quickly intractable. To cope with this problem, pruning and gating eliminate the less likely hypotheses but can too eliminate good clues.
This problem is no more NP-hard as soon as the associations variables are considered as stochastic variables and moreover statistically independent like in the Probabilistic MHT (PMHT). For example in [20] or [7], the algorithm is presented as an incomplete data problem solved by an EM algorithm. The results are then satisfactory when the measurement equation is linear and when the trajectories are deterministic. In [21] the algorithm is extended to the tracking of maneuvering targets with an hidden "model-switch" process controlled by a Markov probability structure. A comparison of the PMHT with the JPDAF is described in a practical two-target scenario in [18], focusing on the mean-square estimation errors and the percentage of lost tracks. However, these algorithms do not cope with non linear state or measurement models and non Gaussian state or measurement noises.
Under such assumptions (stochastic state equation and non linear state or measurement equation, non Gaussian noises), the particle filter, also named bootstrap filter, is particularly adapted. It mainly consists in propagating a weighted set of particles which approximates the probability density of the state conditionally to the observations. It can be applied under very weak hypotheses and is very easy to implement. In particular, the measurement equation can be non linear and the dynamic and observation noises can be non Gaussian. Such filters have been used in very various areas for Bayesian filtering, under different names : the bootstrap filter for target tracking in [9], the Condensation algorithm in computer vision [11]

are two examples among others. At the beginnings of particle filtering, the algorithm was only composed of two periods : the particles were predicted according to the state equation during the prediction step; then their weights were calculated with the likelihood of the new observation combined with the former weights. A resampling step has rapidly been added to dismiss the particles with lower weights and avoid the degeneracy of the particle set with a unique particle of high weight ([9]). Many ways have been developed to accomplish this resampling whose final goal is to enforce particles in areas of high likelihood. The periodicity of this resampling has also been studied. Moreover the use of kernels filters ([10]) has been introduced to regularize the sum of Dirac densities represented by the particles when the dynamic noise of the state equation was too low (see[17]). After all these studies, it seems quite tempting to study the possible extensions of the particle filter to multiple object tracking. In computer vision a probabilistic exclusion principle has been developed in [16] to track multiple objects but the algorithm is very dependent of the observation model and is only applied for two objects. We propose here a quite general algorithm for multitarget tracking in the passive sonar context.
This work is organized as follows. In section II, we describe the basic particle filter with two versions for the resampling step which can be systematic or adaptive. The superiority of adaptive resampling towards systematic resampling is clearly demonstrated in a passive sonar application in section III. Section IV, the central part of this work, deals with an extension of the basic filter to multiple objects. The new algorithm combines the two major steps of prediction and weighting of the classical particle filter with Gibbs sampler computing an estimation of the vector of the assignment probabilities. We introduce two different versions of this Gibbs sampler which take into account the past information in a different way. An application to bearings-only MTT enables us to compare the results obtained with these two versions. In each case, the data association problem is overcomed and the first version is clearly superior. An extension to multi-receiver data in the context of multiple targets ends this section and highlights the good versatility of our approach. As concerns the notational conventions, we always use the index i to refer to one among the M tracked objects. The index j designates one of the m_t observations obtained at instant t. The index n is reserved for the N particles denoted by s. Finally τ is used for indexing the iterations in the Gibbs Sampler and r is used for the different receivers.

2 The basic particle filter

For the sake of completeness, the basic particle filter is now briefly reviewed. The general principle of sampling it relies on will be used throughout the paper. We consider a dynamic system represented by the stochastic process $(X_t) \in \mathbb{R}^{n_x}$ whose temporal evolution is given by the state equation:

$$X_t = F_t(X_{t-1}, V_t). \tag{1}$$

We want to estimate the state vector (X_t) at discrete times with the help of system's observations which are realizations of the stochastic process $(Y_t) \in \mathbb{R}^{n_y}$ governed by the measurement equation:

$$Y_t = H_t(X_t, W_t). \tag{2}$$

The two processes $(V_t) \in \mathbb{R}^{n_v}$ and $(W_t) \in \mathbb{R}^{n_w}$ are only supposed to be independent white noises. Moreover, it is to be noted that no linearity hypothesis on F_t and H_t is done.
We will denote by $Y_{0:t}$ the sequence of the random variables $(Y_0, \dots, Y_t)$ and by $y_{0:t}$ one realization of this sequence.
Our problem consists in computing at each time t the conditional density L_t of the state X_t given all the observations accumulated up to t, i.e., $L_t = p(X_t|Y_0 = y_0, \dots, Y_t = y_t)$ and also in estimating any functional of the state $g(X_t)$ by the expectation $\mathbb{E}(g(X_t)|Y_{0:t})$. The Recursive Bayesian filter, also named Optimal Filter, resolves exactly this problem in two steps at each time t.
Suppose we know L_{t-1}. The *prediction step* is done according to the following equation:

$$p(X_t = x_t|Y_{0:t-1} = y_{0:t-1}) = \int_{\mathbb{R}^{n_x}} p(X_t = x_t|X_{t-1} = x)L_{t-1}(x)dx. \tag{3}$$

Using (1), we can calculate $p(X_t = x_t|X_{t-1} = x)$:

$$\begin{aligned} p(X_t = x_t|X_{t-1} = x) &= \int_{\mathbb{R}^{n_v}} p(X_t = x_t|X_{t-1} = x, V_t = v)p(V_t = v|X_{t-1} = x)dv \quad (4) \\ &= \int_{\mathbb{R}^{n_v}} \delta(x_t - F_t(x, v))p(V_t = v)dv \quad (5) \end{aligned}$$

where $\delta(x)$ denotes the Dirac distribution.
Then, in the *correction step*, the Bayes's rule enables us to compute L_t:

$$L_t(x_t) = \frac{p(Y_t = y_t|X_t = x_t)p(X_t = x_t|Y_{0:t-1} = y_{0:t-1})}{p(Y_t = y_t|Y_{0:t-1} = y_{0:t-1})}. \tag{6}$$

We can rewrite $p(Y_t = y_t|X_t = x_t)$ with the measurement equation (2):

$$p(Y_t = y_t|X_t = x_t) = \int_{\mathbb{R}^{n_w}} \delta(y_t - H_t(x_t, w))p(W_t = w)dw. \tag{7}$$

Moreover, the normalizing denominator $p(Y_t = y_t|Y_{0:t-1} = y_{0:t-1})$ is given by the following integral:

$$p(Y_t = y_t|Y_{0:t-1} = y_{0:t-1}) = \int_{\mathbb{R}^{n_x}} p(Y_t = y_t|X_t = x)p(X_t = x|Y_{0:t-1} = y_{0:t-1})dx. \tag{8}$$

These equations can be applied as written if we assume restrictive hypothesis such as Kalman Filter's ones. The functions F_t and G_t are then supposed to be linear and the noises V_t and W_t to be Gaussian. Unfortunately this modeling is not appropriate in many problems in signal and image processing, which renders the calculations of the integrals in (4) and (8)

infeasible (no closed-form).
The particle filter, named also Condensation algorithm or Sampling Importance Resampling (SIR) algorithm, proposes to approximate the densities $(L_t)_t$ by a finite weighted sum of N Dirac densities centered on elements of $\mathbb{R}^{n_x}$, named particles.
The application of the particle filter requires that one knows how:

- to sample from initial prior marginal $p(X_0)$;
- to sample from $p(V_t)$ for all t;
- to compute $p(Y_t = y_t | X_t = x_t)$ for all t through a known function l_t such that $l_t(y;x) \propto p(Y_t = y | X_t = x)$ where missing normalization must not depend on x.

The first particle set S_0 is created by drawing N independent realizations from $p(X_0)$ and assigning uniform weight $1/N$ to each of them. Then, suppose we dispose at time $t-1$ of the particle set $S_{t-1} = (s_{t-1}^n, q_{t-1}^n)_{n=1,\ldots,N}$ where $\sum_{n=1}^N q_{t-1}^n = 1$. Posteriori marginal L_{t-1} is then estimated by the probability density $L_{S_{t-1}} = \sum_{n=1}^N q_{t-1}^n \delta_{s_{t-1}^n}$.
The *prediction step* consists in propagating each particle of S_{t-1} thanks to the equation evolution (1).
The weight of each particle is updated during the *correction step*. Up to a constant, equation (6) comes down to adjust the weight of predictions by multiplying it by the likelihood $p(y_t|x_t)$. The algorithm is described in Algorithm 1 below.

Algorithm 1 Basic particle filter

Prediction
for $n = 1, \ldots, N$ **do**
  Generate a random sample v_t^n from $p(V_t)$;
  Compute $s_t^n = F_t(s_{t-1}^n, v_t^n)$.
end for
Correction
for $n = 1, \ldots, N$ **do**
  Compute $q_t^n = \frac{l_t(y_t; s_t^n) q_{t-1}^n}{\sum_{n=1}^N l_t(y_t; s_t^n) q_{t-1}^n}$.
end for
Estimation
Estimate $\mathbb{E}g(x_t)$ by $\hat{\mathbb{E}}g(x_t) = \sum_{n=1}^N q_t^n g(s_{t|t-1}^n)$.

By construction, the particles are independent. The weak convergence of the probability density L_{S_t} towards L_t when $N \to \infty$ with rate $1/\sqrt{N}$ can be proved. However, the error is not uniformly bounded in time. The density L_{S_t} is often multi-modal as several hypotheses about the position of the object can be made at one time. It is for instance the case when one object is tracked with an important clutter. Several hypotheses about the object position can then be kept if the set of particles splits into several subsets. This is where the great strength of this filter lies.

The particle sets enables to estimate any functionnal of X_t in particular the two first moments with $g(x) = x$ and $g(x) = x^2$ respectively. The mean can be used to estimate the position of one object but it can be a bad estimator if the law is highly multimodal. In such cases the ideal would be to calculate the mean only over the particles that contribute to the principal mode but such an estimator has not been developed for the moment.
In practice, the particle set is finite and the major drawback of this algorithm is the degeneracy of the particle set : only few particles keep high weights and the others have very small ones. The first carry the information but the second are useless. The resampling is a good way to remedy this drawback because it cancels the particles of smallest weight. The stochastic resampling consists in sampling N particles with replacement in the particles set with the probability q^n to draw s^n. The new particles have uniform weights equal to $1/N$. The resampling can be implemented with a complexity in $O(N)$ in a way detailed in appendix (6.2). The resampling step does not modify the theoretical convergence as the two particle sets follow asymptotically the same density.
A first solution, adopted in [9] for example, consists in applying the resampling step at each time period. The corresponding algorithm of particle filter with systematic resampling is described in Algorithm 2.
To measure the degeneracy of the algorithm, the effective sample size N_{eff} has been introduced in [12], [14] in the more general context of importance resampling. We can estimate this quantity by $\hat{N}_{eff} = 1/\sum_{n=1}^{N}(q_t^n)^2$ which measures the number of meaningful particles. The resampling is then done only if $\hat{N}_{eff} < N_{threshold}$. It enables the particle set to better learn the process and to keep its memory during the interval where no resampling occurs.

Algorithm 2 Particle filter with systematic resampling

Prediction
for $n = 1, \ldots, N$ **do**
Generate a random sample v_t^n from $p(V_t)$;
Compute $s_{t|t-1}^n = F_t(s_{t-1}^n, v_t^n)$.
end for
Correction
for $n = 1, \ldots, N$ **do**
Compute $q_t^n = \frac{l_t(y_t; s_{t|t-1}^n)q_{t-1}^n}{\sum_{n=1}^{N} l_t(y_t; s_{t|t-1}^n)q_{t-1}^n}$.
end for
Estimation
Estimate $\mathbb{E}g(x_t)$ by $\hat{\mathbb{E}}g(x_t) = \sum_{n=1}^{N} q_t^n g(s_{t|t-1}^n)$.
Resampling
for $n = 1, \ldots, N$ **do**
Draw s_t^n from $\sum_{k=0}^{N} q_t^k \delta_{s_{t|t-1}^k}$;
Set $q_t^n = 1/N$.
end for

Details can be found in [12], [14] or [5]. Appendix (6.2) presents the practical way used to resample the particle set.

Algorithm 3 Particle filter with adaptive resampling

Prediction
for $n = 1, \ldots, N$ **do**
  Generate a random sample v_t^n from $p(V_t)$;
  Compute $s_{t|t-1}^n = F_t(s_{t-1}^n, v_t^n)$.
end for
Correction
for $n = 1, \ldots, N$ **do**
  Compute $q_t^n = \frac{l_t(y_t; s_{t|t-1}^n) q_{t-1}^n}{\sum_{n=1}^N l_t(y_t; s_{t|t-1}^n) q_{t-1}^n}$.
end for
Estimation
Estimate $\mathbb{E}g(x_t)$ by $\hat{\mathbb{E}}g(x_t) = \sum_{n=1}^N q_t^n g(s_{t|t-1}^n)$.
Effective size estimation
Calculate $\hat{N}_{eff} = \frac{1}{\sum_{n=1}^N (q_t^n)^2}$.
Resampling
if $\hat{N}_{eff} < N_{threshold}$ **then**

  for $n = 1, \ldots, N$ **do**
    Draw s_t^n from $\sum_{k=0}^N q_t^k \delta_{s_{t|t-1}^k}$;
    Set $q_t^n = 1/N$.
  end for
else
  for $n = 1, \ldots, N$ **do**
    $s_t^n = s_{t|t-1}^n$
  end for
end if

3 Application to bearings-only problems

To illustrate the previous algorithms, we first deal with classical bearings-only problems. The object is then a "point-object" in the $x - y$ plane. In the context of a slowly maneuvering target, we have chosen a nearly-constant-velocity model (see[13] for a review of the principal dynamical models used in this domain).

3.1 The model

The state vector X_t represents the coordinates and the velocities in the $x - y$ plane : $X_t = (x_t, y_t, vx_t, vy_t)$. The discretized state equation associated with time period Δt is:

$$X_{t+\Delta t} = \begin{pmatrix} Id_2 & \Delta t\, Id_2 \\ 0 & Id_2 \end{pmatrix} X_t + \begin{pmatrix} \frac{\Delta t^2}{2} Id_2 \\ \Delta t\, Id_2 \end{pmatrix} V_t, \tag{9}$$

where Id_2 is the identity matrix in dimension 2 and V_t is a Gaussian zero-mean vector of covariance matrix $\Sigma_V = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$. Let $\hat{X}_t$ be the estimation of X_t computed by the particle filters with $g(x) = x$, i.e, . $\hat{X}_t = \sum_{n=1}^{N} q_t^n s_{t|t-1}^n$.
The observations are available at discrete times according to:

$$Y_t = \arctan\left(\frac{x_t - x_t^{obs}}{y_t - y_t^{obs}}\right) + W_t, \tag{10}$$

where W_t is a zero-mean Gaussian noise of covariance σ_w^2 independent of V_t. x_{obs} and y_{obs} are the Cartesian coordinates of the observer, which are known.

3.2 Results of particle filters with systematic and adaptive resampling respectively

The initial position of the target and of the observer are

$$X_0 = \begin{pmatrix} 200 \text{ m} \\ 1500 \text{ m} \\ 1.0 \text{ ms}^{-1} \\ -0.5 \text{ ms}^{-1} \end{pmatrix}; \quad X_0^{obs} = \begin{pmatrix} 200 \text{ m} \\ -3000 \text{ m} \\ 1.2 \text{ ms}^{-1} \\ +0.5 \text{ ms}^{-1} \end{pmatrix}.$$

The observer is following a leg by leg trajectory. Its velocity vector is constant on each leg and modified at the following instants:

$$\begin{pmatrix} vx_{200}^{obs} \\ vy_{200}^{obs} \end{pmatrix} = \begin{pmatrix} -0.6 \\ 0.3 \end{pmatrix}; \begin{pmatrix} vx_{400}^{obs} \\ vy_{400}^{obs} \end{pmatrix} = \begin{pmatrix} 2.0 \\ 0.3 \end{pmatrix}; \begin{pmatrix} vx_{600}^{obs} \\ vy_{600}^{obs} \end{pmatrix} = \begin{pmatrix} -0.6 \\ 0.3 \end{pmatrix}$$

$$\begin{pmatrix} vx^{obs}_{800} \\ vy^{obs}_{800} \end{pmatrix} = \begin{pmatrix} 2.0 \\ 0.3 \end{pmatrix}; \begin{pmatrix} vx^{obs}_{900} \\ vy^{obs}_{900} \end{pmatrix} = \begin{pmatrix} -0.6 \\ 0.3 \end{pmatrix}.$$

The sequence of estimated target bearings are presented in figure 1. They are simulated with a Gaussian noise of variance $\sigma_w^2 = 0.05$ radians (about 3 degrees) every time period, i.e every 6 seconds. The dynamic noise is a normal zero-mean Gaussian vector with $\sigma_x^2 = \sigma_y^2 = 0.001\text{ms}^{-1}$. We use the same dynamic noise to predict the particle.
First, we can calculate the estimated trajectory for one particular run of the particle filter. The true and estimated trajectories obtained with 10000 particles are plotted using systematic resampling in figure 2 and adaptive resampling in figure 3. The threshold $N_{threshold}$ is fixed to 0.9. The average frequency of resampling for such a threshold is 0.132 for $R = 100$ different runs of the filter with adaptive resampling. The values of $\hat{N}_{eff}$ are presented in figure 4 for a particular run.

To assess the accuracy of the two filters, we have computed the bias and standard de-

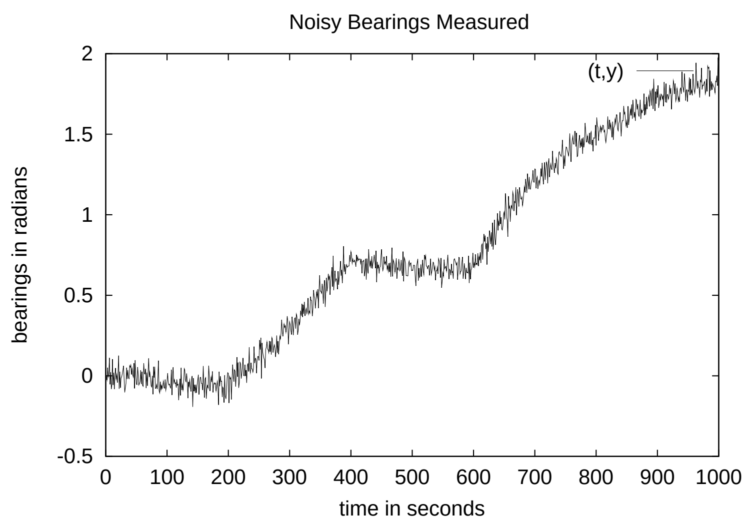


Figure 1: Bearings measured at each time period for the single target bearings-only experiment

viation with $R = 100$ different runs of each filter for the four components of the vector X. Table 1 contains the results. At each time t the bias for the l^{th} component of X is defined by

$$bias_{t,l} = \frac{1}{R}\sum_{r=1}^{R}(\hat{X^r_{t,l}} - X_{t,l}), \tag{11}$$

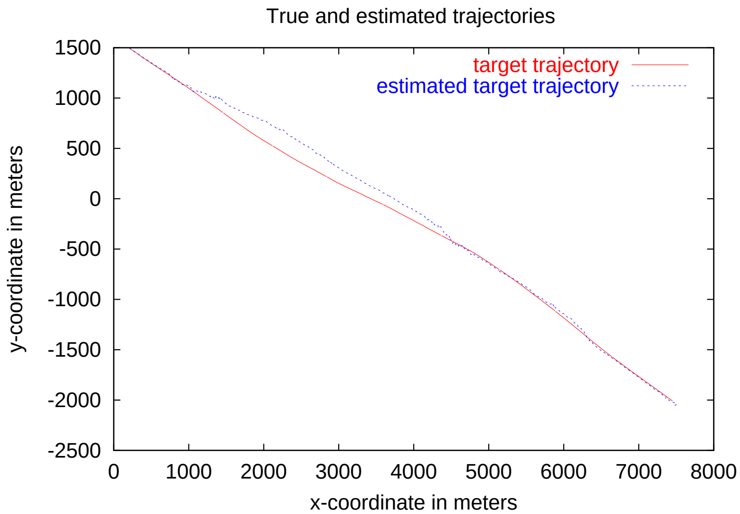


Figure 2: True and estimated trajectories obtained with 10000 particles and systematic resampling

and the standard deviation:

$$std_{t,l} = \frac{1}{R}\sum_{r=1}^{R} \hat{X^r_{t,l}}^2 - (\frac{1}{R}\sum_{r=1}^{R} \hat{X^r_{t,l}})^2. \tag{12}$$

To avoid compensation of elementary bias (of opposite signs), we average the absolute values of the bias $bias_{t,l}$. Then, we define

$$bias_l = \frac{1}{T}\sum_{t=1}^{T} |bias_{t,l}|, \tag{13}$$

and we average the standard deviations

$$std_l = \frac{1}{T}\sum_{t=1}^{T} std_{t,l}. \tag{14}$$

The differences between the biases are not very significant (still, it is slightly lower with adaptive resampling), but the standard deviation is significantly higher in the case of systematic resampling. The estimates provided by one particular run are then less reliable with systematic resampling than with adaptive one. Adaptive resampling will thus be used in

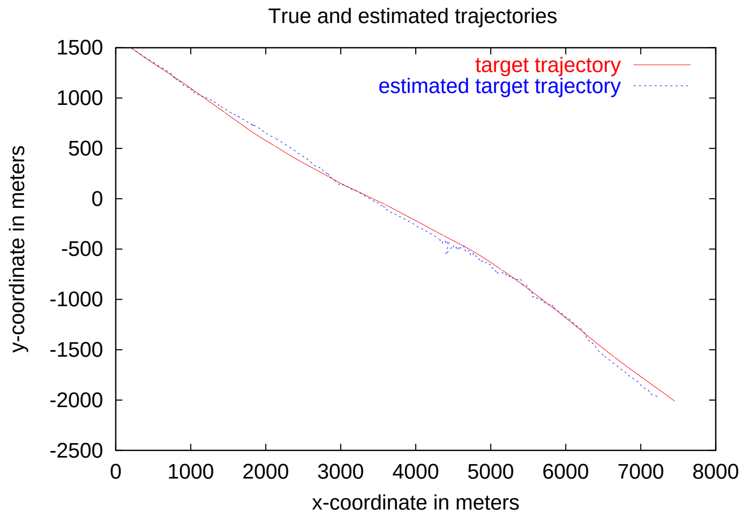


Figure 3: True and estimated trajectories obtained with 10000 particles and adaptive resampling

Table 1: Comparison between particle filtering performance with systematic and adaptive resampling, in the single-object bearings-only synthetic example

	Bias			Std		
	systematic	adaptive	systematic/adaptive	systematic	adaptive	systematic/adaptive
x	91.153	83.656	1.089	1.902	0.799	2.37
y	61.05	58.507	1.043	0.9714	0.4018	2.47
v_x	0.0879	0.0859	1.023	0.00098	0.000398	2.46
v_y	0.07733	0.07318	1.05	0.00077	0.00034	2.26

the rest of this work.

It is to be noted that in the case of real data the variance of the state noise does not need to be known precisely : if it is overestimated, the particles are predicted in a larger area but are also more eliminated in the resampling step. Practically it is a great strength of particle filtering.

The purpose of this section was to provide a general representation of particle filtering. We can now turn to its extension to multi-target tracking.

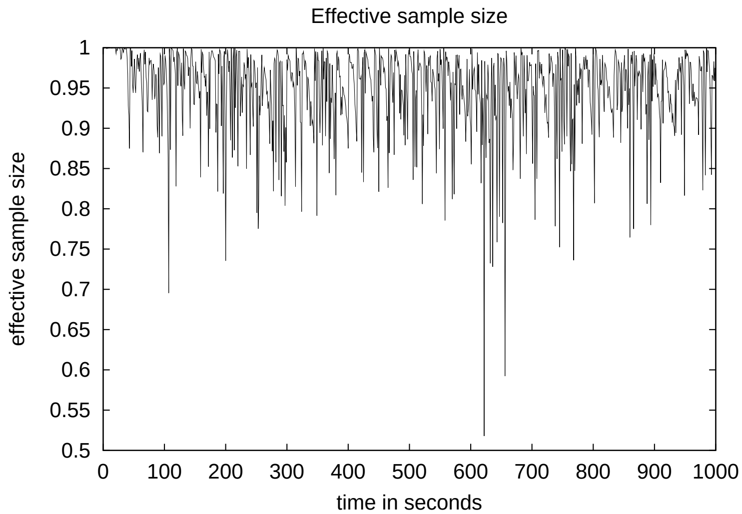


Figure 4: The values of the effective sample size $\hat{N}_{eff}$ obtained with 10000 particles and adaptive resampling

4 Multitarget particle filters

4.1 Notations

Let M be the number of objects to track (assumed to be known and fixed). The state vector we have to estimate is made by concatenating the state vector of each object. At time t, $X_t = (X_t^1, \ldots, X_t^M)$ follows the state equation (1) decomposed in M partial equations:

$$X_t^i = F_t^i(X_{t-1}^i, V_t^i) \qquad \forall\, i = 1, \ldots, M. \tag{15}$$

The noises (V_t^i) and $(V_t^{i'})$ are only supposed to be white both temporally and spatially, independent for $i \neq i'$.
The observation vector at time t is denoted by $y_t = (y_t^1, \ldots, y_t^{m_t})$. Following the seminal ideas of R. Streit and T. Luginbuhl [20], we introduce the stochastic vector $K_t \in \{1, \ldots, M\}^{m_t}$ such that $K_t^j = i$ if y_t^j is issued from the i^{th} object. In this case, y_t^j is a realization of the stochastic process:

$$Y_t^j = H_t^i(X_t^i, W_t^j) \text{ if } K_t^j = i. \tag{16}$$

Again, the noises (W_t^j) and $(W_t^{j'})$ are only supposed to be white noises, independent for $j \neq j'$. We assume that the functions H_t^i are such that they can be associated to functional forms l_t^i defined by $l_t^i(y; x) \propto p(Y_t^j = y | K_t^j = i, X_t^i = x)$.

We make the assumption that one measurement can originate from one object or from the clutter and that one object can produce zero or several measurements at one time. For that, we dedicate the model 0 to false alarms. The false alarms are supposed to be uniformly distributed in the observation area. Their number is assumed to arise from a Poisson density of parameter λV where V is the volume of the surveillance area and λ the number of false alarms by volume unity. Of course, we do not associate any kinematic model to false alarms and then no particles represent their density. Let $\pi_t \in [0, 1]^{M+1}$ defined by $\pi_t^i = \mathbb{P}(K_t^j = i)$ for all $j = 1, \dots m_t$. This definition implicitly assumes that the probabilities π_t^i are independent of the measurements as their indexation is arbitrary. These assumptions imply that m_t may differ from M and that the association is exclusive and exhaustive. In particular, $\sum_{i=0}^{M} \pi_t^i = 1$. Furthermore, it is assumed that the assignment vector K_t has independent components (see[20] and [21]).

Let us first study the joint density $p(X_t, Y_t, \Pi_t, K_t)$ (implicitly conditionally to X_{t-1}).

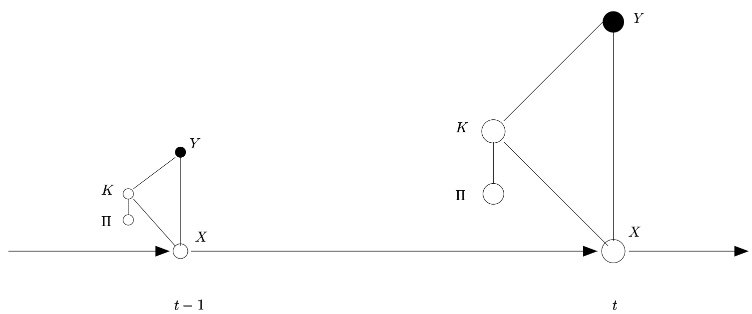


Figure 5: Independence graph of the X, Y, K, Π stochastic variables

We will drop the subscript t for a few lines.

$$\begin{aligned} p(X, Y, \Pi, K) &= p(Y|K, \Pi, X)p(K|\Pi, X)p(\Pi, X) && (17) \\ &= p(Y|K, X) \quad p(K|\Pi) \quad p(\Pi)p(X) && (18) \end{aligned}$$

where the last equality is based on the following independence assumptions

- $p(Y|K, X) = \prod_j p(Y^j|K^j, X) = \prod_j \begin{cases} l^i(y^j|x^i) & \text{with } i = K^j \in \{1 \dots M\}; \\ 1/V & \text{if } K^j = 0. \end{cases}$
- $p(K|\Pi) = Multinomial(n(K)|m_t, \Pi)$ with $n^i(K) \triangleq \sharp\{j : K^j = i\}$, i.e., the vector $(n^0(K), \dots, n^M(K))$ follows a multinomial law of size m_t and of parameters $(\Pi^0, \dots, \Pi^M)$;

- $p(\Pi) = p(\Pi_0)p(\Pi_1, \ldots, \Pi_M|\Pi_0)$ where $p(\Pi_1, \ldots, \Pi_M|\Pi_0)$ is uniform on the $[0, 1-\pi_0]^M$'s hyperplane $\sum_{i=1}^M \pi_i = 1 - \pi_0$. π_0 is a constant that can be computed:

$$\pi_0 = \mathbb{P}(K_t^j = 0) \tag{19}$$

$$= \sum_{l=0}^{m_t} \mathbb{P}(K_t^j = 0|N_t^0 = l)\mathbb{P}(N_t^0 = l) \tag{20}$$

$$= \sum_{l=0}^{m_t} \frac{l}{m_t} \exp -\lambda V \frac{(\lambda V)^l}{l!} \tag{21}$$

- $p(X)$, i.e., $p(X_t|X_{t-1})$ conditionally to X_{t-1}.

This independence structure of this distribution is illustrated in figure 5.
The density L_t is now multimodal in a global sense, each principal mode corresponding to one object. To represent this density by a particle set we could use particles of X_t^i's dimension and distribute them among the objects (providing that the objects have the same dimension which might not be the case for complex objects in image processing). Each of them would then be known to represent a given object. The weight of particle n is then:

$$q_t^n \propto p(Y_t = (y_t^1, \ldots, y_t^{m_t})|X_t^i = s_t^n) \tag{22}$$

As the observations cannot be supposed independent conditionally to only one of the M objects, the likelihood cannot be decomposed in the product of likelihoods of each observation. One solution could be to use Bayes's rule and to compute $p(X_t^i = s_t^n|Y_t = (y_t^1, \ldots, y_t^{m_t}))$, but this requires that every object produces at less one measure (to be able to use the Total Probability Theorem), which seems too restrictive. Moreover, with such a representation the resampling step could disadvantage one object if all "its" particles have low weights. This object would be no more represented in the likely event all "its" particles are discarded after the resampling. For these reasons, we instead use particles whose dimension is the sum of the ones of the individual state spaces corresponding to each object (e.g. $4M$ for bearings-only example). Each of these concatenated vectors then gives jointly a representation of all objects. We will call the filters associated to this representation the joint filters.

4.2 Joint filters

The initial particle set $S_0 = (s_0^n, 1/N)_{n=1,\ldots,N}$ is such that each component $s_0^{n,i}$ for $i = 1, \ldots, M$ is sampled from $p(X_0^i)$ independently from the others. Assume we have obtained $S_{t-1} = (s_{t-1}^n, q_{t-1}^n)_{n=1,\ldots,N}$ with $\sum_{n=1}^N q_{t-1}^n = 1$. Each particle is a vector of dimension $\sum_{i=1}^M n_x^i$ where we denote by $s_{t-1}^{n,i}$ the i^{th} component of s_{t-1}^n and where n_x^i designates the dimension of object i.

The prediction can be done according to the following equation:

$$\text{For } n=1,\dots,N \; s^n_{t|t-1} = \begin{pmatrix} F^1_t(s^{n,1}_{t-1}, v^{n,1}_t) \\ \vdots \\ \vdots \\ F^M_t(s^{n,M}_{t-1}, v^{n,M}_t) \end{pmatrix} \tag{23}$$

Examine now the computation of the likelihood of the observations conditioned by the n^{th} particle. We can write for all $n=1,\dots,N$:

$$\begin{aligned} p(Y_t = (y^1_t,\dots,y^{m_t}_t)|X_t = s^n_{t|t-1}) &= \prod_{j=1}^{m_t} p(y^j_t|s^n_{t|t-1}), && (24) \\ &\propto \prod_{j=1}^{m_t} [\frac{\pi^0_t}{V} + \sum_{i=1}^{M} l^i_t(y^j_t; s^{n,i}_{t|t-1})\pi^i_t]. && (25) \end{aligned}$$

It must be noted that equation (24) is true only under the assumption of conditional independence of the measures, which we will make. Moreover, the normalization factors between l^i_t and $p(Y^j_t = y|K^j_t = i, X^i_t = x)$ must be the same for all i to write the last equality (25).
It remains to estimate the association probabilities $(\pi^i_t)_{i=1,\dots,M}$ which can be seen as the stochastic coefficients of the $M-$component mixture. Let $y^1,\dots,y^{m_t}$ be independent observations from a mixture density with M components, where M is finite but can be unknown. The missing data, denoted k^j, indicates us which component each observation y^j belongs to. Two main ways can be found in the literature to estimate the parameters of this model : the EM method (and its stochastic version, the SEM algorithm [3]) and the Data Augmentation method. The second one is in fact a Gibbs sampler.
In [7],[20],[21], the EM algorithm is extended and applied to multitarget tracking. This method implies that the vectors π_t and X_t are considered as parameters to estimate. The maximization step can be easily computed in the case of deterministic trajectories and the additionnal use of a MAP estimator enables to achieve it for non deterministic trajectories. Yet, the non linearity of the state and observation functions makes this step very difficult. Moreover, the algorithm supplies the estimations of π_t and X_t whereas we only need the estimate of π_t as the estimation of X_t is the task of the particle filter. Finally, the estimation is done iteratively in a batch approach which we would like to avoid in our approach with particle filter. Then, the EM algorithm is not really appropriate to particle filtering.
The Data Augmentation algorithm is quite different in its principle. The vectors X_t, K_t and π_t are considered as random variables and prior densities must be put on them. Such methods have been studied in [4] or [19] for instance. It can be run sequentially at each time period. Let us recall the main ideas of this algorithm in the general context of mixture model. For $\theta = (X_t, K_t, \Pi_t)$, it consists in generating a Markov chain which converges towards the stationary distribution $p(\theta|Y_{0:t})$ which cannot be sampled directly. For that, we must be able to get a partition $\theta^1,\dots,\theta^P$ of θ and to sample alternatively from the

conditional posterior distribution of each component of the partition. Assume the τ first elements of the Markov chain $(\theta_1, \dots, \theta_\tau)$ have been drawn. We sample the P components of $\theta_{\tau+1}$ as follows:

$$\begin{array}{l}
\texttt{Draw } \theta^1_{\tau+1} \texttt{ from } p(\theta^1|Y_{0:t}, \theta^2_\tau, \dots, \theta^P_\tau) \\
\texttt{Draw } \theta^2_{\tau+1} \texttt{ from } p(\theta^2|Y_{0:t}, \theta^1_{\tau+1}, \theta^3_\tau, \dots, \theta^P_\tau) \\
\qquad\quad \vdots \qquad\qquad \vdots \\
\texttt{Draw } \theta^P_{\tau+1} \texttt{ from } p(\theta^P|Y_{0:t}, \theta^1_{\tau+1}, \dots, \theta^{P-1}_{\tau+1})
\end{array}$$

The proof of the convergence of the Markov chain $(\theta_\tau)_\tau$ is outlined in Appendix (6.1). We propose to use it in the particle filter extended to multiple targets.
In our case, at a given instant t, we follow this algorithm with

$$\left\{ \begin{array}{ll} \theta^j = K^j_t & \text{for } j = 1, \dots, m_t; \\ \theta^{m_t+i} = \pi^i_t & \text{for } i = 1, \dots, M; \\ \theta^{m_t+M+i} = X_t & \text{for } i = 1, \dots, M. \end{array} \right. \tag{26}$$

The initialization of the algorithm consists in assigning uniform association probabilities, i.e., $\pi^i_{t,0} = \frac{1-\pi_0}{M}$ for all $i = 1, \dots, M$, and taking $X_{t,0} = \sum_{n=1}^N q^n_{t-1} s^n_{t|t-1}$, i.e., the centroid of the predicted particle set. Then, suppose that at instant t we have already simulated $(\theta_{t,1}, \dots, \theta_{t,\tau})$. The $\tau + 1^{th}$ iteration is handled as follows.

- As the $(K^j_t)_{j=1,\dots,m_t}$ are supposed to be independent, their individual conditional density reads:

$$p(K^j_t|Y_{0:t}, X_t, (K^l_t)_{l \neq j}, \Pi_t) = p(K^j_t|Y^j_t, X_t, \Pi_t). \tag{27}$$

  (K^j_t) are discrete variables and we can write:

$$\mathbb{P}(K^j_t = i|Y^j_t = y^j_t, X_t, \Pi_t) = \frac{p(Y^j_t = y^j_t|K^j_t = i, X_t, \Pi_t)\mathbb{P}(K^j_t = i|X_t, \Pi_t)}{p(Y^j_t = y^j_t|X_t, \Pi_t)} \tag{28}$$

$$\propto \left\{ \begin{array}{ll} \pi^i_t l^i_t(y^j_t; x^i_t) & \text{if } i = 1, \dots, M. \\ \pi^0_t / V & \text{if } i = 0 \end{array} \right. \tag{29}$$

  The realizations $k^j_{t,\tau+1}$ of the vector $K_{t,\tau+1}$ are then sampled according to the weights $p^{j,0}_{t,\tau+1} = \pi^0_t/V, p^{j,i}_{t,\tau+1} = \pi^i_{t,\tau} l^i_t(y^j_t; x^i_{t,\tau})$ for $i = 1, \dots, M$. The practical way the K^j_t are sampled from their discrete law is explicited in appendix (6.2).

- Mixture proportion vector $\Pi^{1:M}_{t,\tau+1}$ is drawn from the conditional density:

$$p(\Pi^{1:M}_t|K_{t,\tau+1}, X_{t,\tau}, Y_{0:t}) = p(\Pi^1_t, \dots, \Pi^M_t|K^1_{t,\tau+1}, \dots, K^M_{t,\tau+1}, X_{t,\tau}, Y_{0:t}) \tag{30}$$

$$\propto p(K^1_{t,\tau+1}, \dots, K^M_{t,\tau+1}|\Pi^1_t, \dots, \Pi^M_t) p(\Pi^1_t, \dots, \Pi^M_t) \tag{31}$$

$$= (1 - \pi_0) Dirichlet(\Pi_t|M, \{n^i(K_{t,\tau+1})_{i=1,\dots,M}\}) \tag{32}$$

  where we denote by $n^i(K)$ the number of k^j equal to i.

- $X_{t,\tau+1}$ has to be sampled according to the density

$$p(X_t|Y_{0:t}, K_{t,\tau+1}, \Pi_{t,\tau+1}) = \prod_{i=1}^{M} p(X_t^i|Y_{0:t}, K_{t,\tau+1}, \Pi_{t,\tau+1}) \qquad (33)$$

The values of $K_{t,\tau+1}$ can imply that one object is associated with zero or several measurements that is why we decompose the preceding product in two products:

$$\prod_{i/\exists j^1,\dots,j^i/K^j_{t,\tau+1}=i} p(X_t^i|Y_{0:t-1}, y_t^{j^1}, \dots, y_t^{j^i}, \Pi_{t,\tau+1}) \prod_{i/\forall j K^j_{t,\tau+1}\neq i} p(X_t^i|Y_{0:t-1}, \Pi_{t,\tau+1}) \qquad (34)$$

- Let i be an integer in the first product. We propose two approaches to sample $X_{t,\tau+1}$:

  ⋆ Without making any additional assumption we can write

$$p(X_t^i|Y_{0:t-1}, y_t^{j^1}, \dots, y_t^{j^i}, \Pi_{t,\tau+1}) = \frac{p(y_t^{j^1}, \dots, y_t^{j^i}|X_t^i)p(X_t^i|Y_{0:t-1})}{p(y_t^{j^1}, \dots, y_t^{j^i}|Y_{0:t-1})} \qquad (35)$$

We are not able to sample directly from the density $\frac{p(y_t^{j^1},\dots,y_t^{j^i}|X_t^i)p(X_t^i|Y_{0:t-1})}{p(y_t^{j^1},\dots,y_t^{j^i}|Y_{0:t-1})}$, for the same reasons as those exposed in section 2 to justify the use of the particle filter (intractability of the integrals). A first solution consists in building the particle set $\Sigma_{\tau+1} = (\sigma^n_{\tau+1}, \chi^n_{\tau+1})_{n=1,\dots,N}$ whose weights $\chi^n_{\tau+1}$ measure the likelihood of the observations affected by $K_{t,\tau+1}$ to object X_t^i . More precisely, we let:

$$\begin{cases} \sigma^n_{\tau+1} = s^{n,i}_{t|t-1}; \\ \chi^n_{\tau+1} = \frac{p(y_t^{j^1},\dots,y_t^{j^i}|X_t^i=\sigma^n_{\tau+1})\, q^n_{t-1}}{\sum_{n=1}^N p(y_t^{j^1},\dots,y_t^{j^i}|X_t^i=\sigma^n_{\tau+1})\, q^n_{t-1}}. \end{cases} \qquad (36)$$

The density $\Lambda_{\tau+1} = \sum_{n=1}^N \chi^n_{\tau+1}\delta_{\sigma^n_{\tau+1}}$ converges weakly towards the density $p(X_t^i|y_t^{j^1}, \dots, y_t^{j^i}, Y_{0:t-1})$. Not being able to sample from this last density, $X^i_{t,\tau+1}$ is drawn as a realization from $\Lambda_{\tau+1}$.

  ⋆ The second solution assumes the measurement equation enables us to sample from the density $P(X_t = x|Y_t = y)$ and to forget the observations from the past. The likelihood $p(X_t^i|Y_{0:t-1}, y_t^{j^1}, \dots, y_t^{j^i}, \Pi_{t,\tau+1})$ is then reduced to $p(X_t^i|y_t^{j^1}, \dots, y_t^{j^i})$. We do not assume that the observations $y_t^{j^1}, \dots, y_t^{j^i}$ are independent but we use their centroid $\bar{y}$ and replace $p(X_t^i = x|y_t^{j^1}, \dots, y_t^{j^i}, K_t^{j^1} = i, \dots, K_t^{j^i} = i)$ by $p(X_t^i = x|\bar{y})$.
  As far as the complexity of these two solutions is concerned, it is to be noticed that the first one depends linearly on the total number of particles whereas the second is independent of it. On the other hand, the second solution requires the ability to sample from $p(X_t|Y_t)$.

- Now let i be an integer in the second product. As we do not have any measure to correct the predicted particles we draw a realization from the density $\sum_{n=1}^{N} q_{t-1}^{n} \delta_{s_{t|t-1}^{n}}$ for $X_{t,\tau+1}^{i}$.

After a finite number of iterations, we estimate the vector π_t by the average of its last realizations:

$$\hat{\pi}_t^i = \frac{1}{\tau_{beg} - \tau_{end}} \sum_{\tau=\tau_{beg}}^{\tau_{end}} \pi_{t,\tau}^i. \tag{37}$$

Finally the weights can be computed according to (24) using the estimation $\hat{\pi}_t^i$ of π_t^i. Like in the case of a unique object a resampling step can be conducted in an systematic or adaptive way. The various forms of the algorithms are summarized in figures 6 and 7.

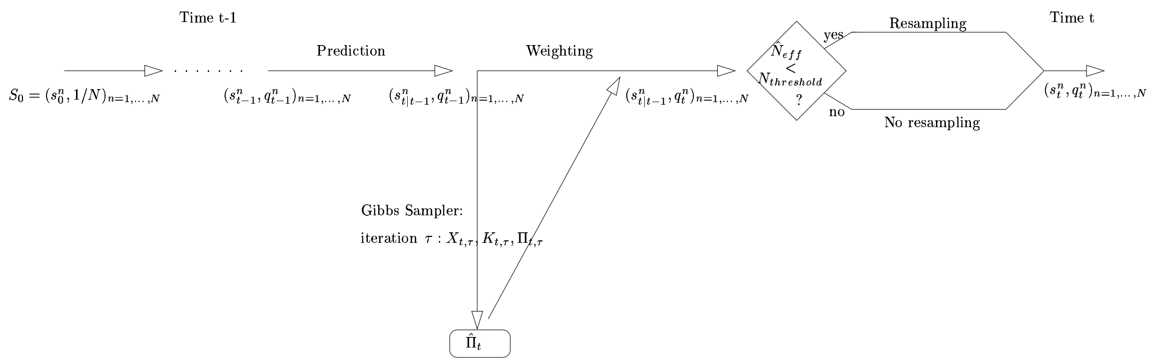


Figure 6: Particle Filter for Multiple Objects with adaptive resampling

- Initialization: $\begin{cases} s_0^n \sim p(X_0) \\ q_0^n = 1/N \end{cases} \quad n = 1, \dots, N.$
- For $t = 1, \dots, T$:
  - Prediction: $\begin{cases} v_t^n \sim p(V_t) \\ s_{t|t-1}^n = F_t(s_{t-1}^n, v_t^n) \end{cases} \quad n = 1, \dots, N.$
  - Weighting
    - ▲ Initialization of the Gibbs sampler: $\begin{cases} \pi_{t,0}^i = \frac{1-\pi_t^0}{M} & i = 1, \dots, M \\ X_{t,0}^i = \sum_{n=1}^N q_{t-1}^n s_{t|t-1}^{n,i} & i = 1, \dots, M \end{cases}$
    - ▲ For $\tau = 0, \dots, \tau_{end}$:
      - △ $K_{t,\tau+1}^j \sim \; p(K_{t,\tau+1}^j = i) \propto \begin{cases} \pi_{t,\tau}^i l_t^i(y_t^j; x_{t,\tau}^i) & \text{if } i = 1, \dots, M; \\ \pi_t^0/V & \text{if } i = 0. \end{cases}$
      - △ $\pi_{t,\tau+1}^{1:M} \sim \mathcal{D}((1 + n^i(K_{t,\tau+1}))_{i=1,\dots,M})$, $n^i(K) \triangleq \sharp\{j : K^j = i\}$.
      - △ For each i such that $\exists j^1, \dots, j^i / K_{t,\tau+1}^{j^l} = i$,

        First solution:

        ▶ $\begin{cases} \sigma_{\tau+1}^n = s_{t|t-1}^{n,i} \\ \chi_{\tau+1}^n = \frac{p(y_t^{j^1}, \dots, y_t^{j^i} | X_t^i = \sigma_{\tau+1}^n)\, q_{t-1}^n}{\sum_{n=1}^N p(y_t^{j^1}, \dots, y_t^{j^i} | X_t^i = \sigma_{\tau+1}^n)\, q_{t-1}^n} \end{cases} \quad n = 1, \dots, N.$

        ▶ $X_{t,\tau+1}^i \sim \sum_{n=1}^N \chi_{\tau+1}^n \delta_{\sigma_{\tau+1}^n}$.

        Second solution:

        ◀ $\bar{y} =$ centroid of $y_t^{j^1}, \dots, y_t^{j^i}$.

        ◀ $X_{t,\tau+1}^i \sim p(X_t|\bar{y})$.
      - △ For each i such that $\nexists j / K_t^j = i$, $X_{t,\tau+1}^i \sim \sum_{n=1}^N q_{t-1}^n \delta_{s_{t|t-1}^{n,i}}$.
    - ▲ $\hat{\pi}_t^i = \frac{1}{\tau_{beg} - \tau_{end}} \sum_{\tau=\tau_{beg}}^{\tau_{end}} \pi_{t,\tau}^i \quad i = 1, \dots, M.$
    - ▲ $q_t^n \propto q_{t-1}^n \prod_{j=1}^{m_t} [\frac{\pi_t^0}{V} + \sum_{i=1}^M l_t^i(y_t^j; s_{t|t-1}^{n,i}) \hat{\pi}_t^i] \quad n = 1, \dots, N.$
  - Return $\widehat{\mathbb{E}} g(X_t) = \sum_{n=1}^N q_t^n g(s_{t|t-1}^n)$.
  - $\hat{N}_{eff} = \frac{1}{\sum_{n=1}^N (q_t^n)^2}$.
  - Resampling : if $\hat{N}_{eff} < N_{threshold}$: $\begin{cases} s_t^n \sim \sum_{k=1}^N q_t^k \delta_{s_{t|t-1}^k} \\ q_t^n = 1/N \end{cases} \quad n = 1, \dots, N.$

Figure 7: *Particle filter algorithm for multiple objects with adaptive resampling.*

4.3 Application to bearings-only problems

The following multitarget scenario has been considered for illustrating our algorithm. Three targets follow a near-constant-velocity model defined by equation (9). They produce one measurement at each time period according to equation (10) except during the time interval [600 700] where the first object does not produce any measurement and the second produces two y^1 and y^2 according to equation (10). The trajectories of the three targets and of the observer are plotted in figure 8 and the differences between the three couples of bearings simulated are plotted in figure 9.

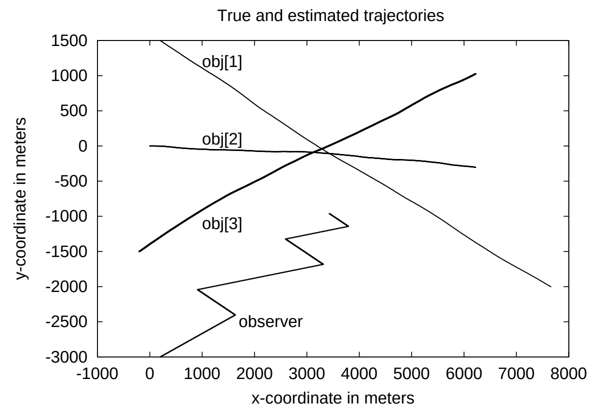
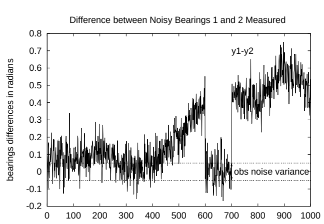
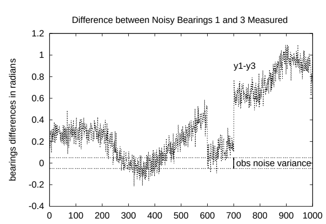


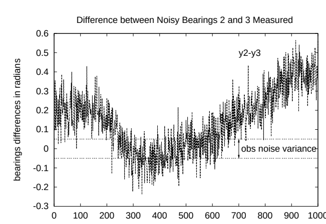
Figure 8: Trajectories of the three targets and of the observer



1



2



3

Figure 9: Differences between the three couples of target bearings at each time period compared to the variance of the observation noise: (1) measurements 1 and 2 (2) 1 and 3 (3) 2 and 3

As soon as the difference between two bearings issued from two different targets is lower than the variance of the observation noise, the two measures cannot be distinguished, which makes this scenario very difficult. This difficulty is increased by the detection hole for the first object. We compare the estimated trajectories in case where the assignment probabilities are respectively:

(α) estimated by the first version of the Gibbs sampler described above

(β) estimated by the second version of the Gibbs sampler described above

Two particular runs of the particle filter with 5000 particles are presented in figure 10.1 and 10.2 using respectively the first and the second version of the Gibbs sampler to estimate the vector π_t. The three estimated trajectories plots show that the data association is overcomed in the two versions of the algorithm. There is no trajectories reversal and the estimations are quite satisfactory for the two versions.

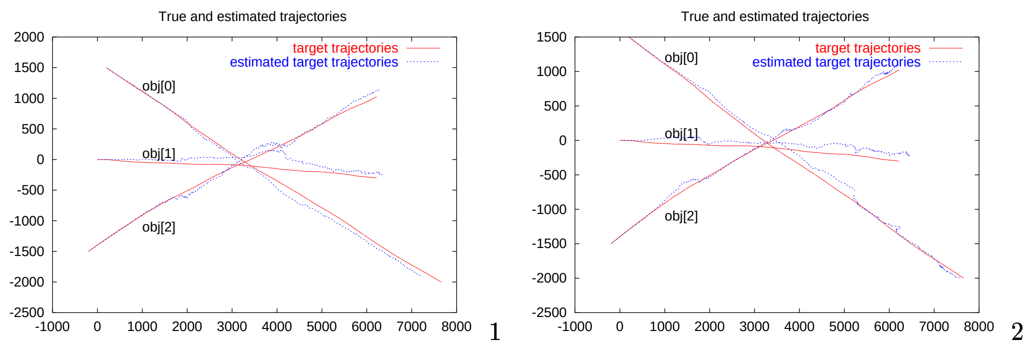


Figure 10: The target trajectories and their estimation with 5000 particles. 1 : estimation of π_t with the first version of the Gibbs sampler 2 : estimation of π_t with the second version of the Gibbs sampler

The computation of the bias and of the variance has been made for 20 runs in each situation (first or second version) with 5000 particles and adpative resampling. These results are reported in table 2. They are better for the first version except for some components of the third object.
Morever the estimation of the π_t is more satisfactory for the first version. Figures 11 and 12 show the results of the estimation of the three components of π_t, respectively with the first and the second versions of the algorithm. Figure 13 represents the average of each component π_t^i over successive intervals of 100 time steps and over the 20 trials. When there is an ambiguity about the origin of the measurements (i.e., when the differences between the bearings are lower than the variance noise), the first version makes π varying in average around 1/3 for $M = 3$ objects and it stabilizes at uniform estimation (1/3 for $M = 3$ objects) when the ambiguity dissapears. By contrast, for the second version the averages of the variations are different according to the components and no stability is reached where

there should be no ambiguity. The momentary measurement gap for the first object is correctly handled by both versions as the first component π_t^1 is instantaneously estimated as 0.15 from instant 600 to 700. However, the second version overestimates the third component during this interval and the double measurement production of the second object does not affect the second component π_t^2. This period is better treated by the first version with an important increase of the second component to 0.44 and a less important one for the third one (0.4). The expected superiority of the first algorithm seems to hold in practice, with a better estimation of π_t.

Table 2: Comparison between the bias and the std on the estimation of the hidden states (x, y, v_x, v_y) for the two versions of the Π_t estimation (columns α and β resp.) with 5000 particles

	bias			std		
π estimation	α	β	β/α	α	β	β/α
object 1 : x	173	250	1.4	6.17	9.84	1.59
y	92.1	137	1.48	4.34	5.9	1.36
v_x	0.138	0.164	1.19	0.0031	0.00477	1.54
v_y	0.11	0.144	1.31	0.00323	0.00424	1.31
object 2 : x	102	127	1.25	4.96	7.34	1.48
y	112	141	1.26	6.51	7.99	1.23
v_x	0.07	0.0880	1.26	0.0026	0.00388	1.49
v_y	0.069	0.0803	1.16	0.00303	0.00378	1.25
object 3 : x	125	116	0.93	4.11	3.63	0.88
y	127	115	0.91	4.77	4.66	0.98
v_x	0.09	0.097	1.08	0.0023	0.00267	1.16
v_y	0.09	0.0903	1.003	0.0027	0.0027	1

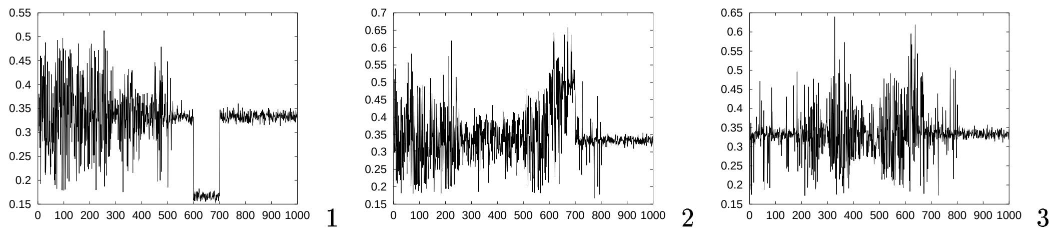


Figure 11: The estimated components of the vector π_t obtained with the first version of the Gibbs sampler and 5000 particles: (1) $\hat{\pi}_t^1$, (2) $\hat{\pi}_t^2$, (3) $\hat{\pi}_t^3$

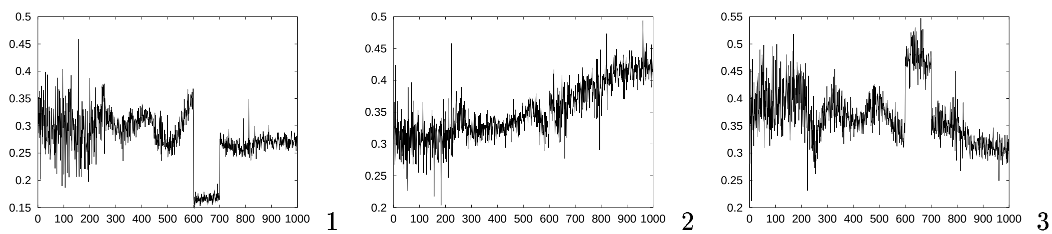


Figure 12: The estimated components of the vector π_t obtained with the second version of the Gibbs sampler and 5000 particles: (1) $\hat{\pi}_t^1$, (2) $\hat{\pi}_t^2$, (3) $\hat{\pi}_t^3$

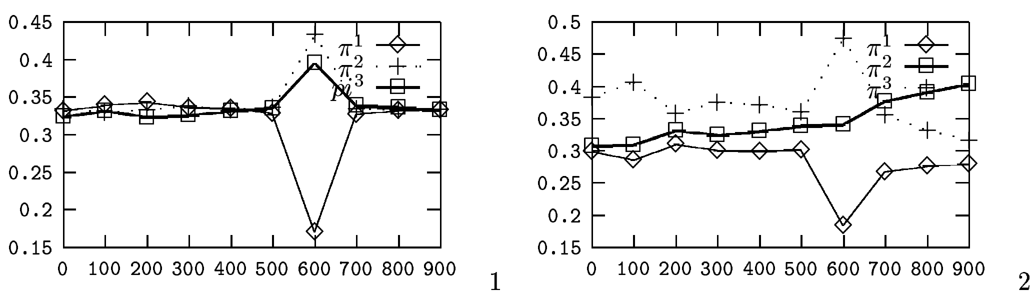


Figure 13: The average of the estimated components of the vector π_t over the consecutive ten time intervals of length 100 and over 20 trials: (1) first version (2) second version of the Gibbs sampler

4.4 Extension to multireceiver data

A natural extension is to consider that observations can be issued from multiple receivers. Let R be their number. We will see that we can easily adapt the particle filter to this situation. We always consider that the M objects obey the state equation (15). Some useful notations must be added to modify the measurement equations. The observation vector at time t will be denoted by $y_t = (y^1_{t,r^1}, \dots, y^{m_t}_{t,r^{m_t}})$ where r^j refers to the receiver which received the j^{th} measure. This measure is then a realisation of the stochastic process

$$Y^j_{t,r^j} = H^i_{t,r^j}(X^i_t, W^j_t) \text{ if } K^j_t = i. \tag{38}$$

We assume the independence of the observations issued from the different receivers. We denote by $l^i_{t,r^j}(y;x)$ the functions which are proportionnal to $p(Y^j_{t,r^j} = y | K^j_t = i, X^i_t = x)$. The likelihood of the observations conditionned by the n^{th} particle is readily obtained:

$$\begin{aligned} p(Y_t = (y^1_{t,r^1}, \dots, y^{m_t}_{t,r^{m_t}}) | X_t = s^n_t) &= \prod_{j=1}^{m_t} p(y^j_{t,r^j} | s^n_t) & (39) \\ &= \prod_{j=1}^{m_t} [\frac{\pi^0_t}{V} + \sum_{i=1}^{M} l^i_{t,r^j}(y^j_{t,r^j}; s^{n,i}_t)\pi^i_t]. & (40) \end{aligned}$$

There is no strong limitation on the use of the particle filter for multireceiver and multitarget tracking. Moreover it can deal with measurements of various periodicities.

5 Conclusion

The multitarget tracking has been investigated in the framework of particle filtering and Gibbs sampling. Target state vectors and association probabilities are estimated jointly without enumeration, pruning or gating, by means of particle sets representing the joint *a posteriori* law of the target states. Various versions, derived from a common formalism, have been considered. This framework is sufficiently versatile to handle a wide variety of situations like multitarget tracking for multireceivers, including non-linear models.

6 Appendix

6.1 Gibbs sampler

By construction, $(\theta_\tau)_{\tau \in \mathbb{N}}$ is a Markov chain. Let us prove it admits the stationary distribution $\lambda = p(\theta | Y_{0:t})$. For that, let us prove that if θ_τ is distributed according to λ then $\theta_{\tau+1}$ is also distributed according to λ. Assume $\theta_\tau = (\theta^1_\tau, \dots, \theta^P_\tau) \sim \lambda$. Then, for all $(z^1, \dots, z^P)$

$$\begin{aligned} p(\theta^1_{\tau+1} = z^1, \theta^2_\tau = z^2, \dots, \theta^P_\tau = z^P) & \\ &= p(\theta^1_{\tau+1} = z^1 | \theta^2_\tau = z^2, \dots, \theta^P_\tau = z^P)\, p(\theta^2_\tau = z^2, \dots, \theta^P_\tau = z^P) \\ &= p(\theta^1 = z^1 | Y_{0:t}, \theta^2 = z^2, \dots, \theta^P = z^P)\, p(\theta^2 = z^2, \dots, \theta^P = z^P | Y_{0:t}) \\ &= p(\theta^1 = z^1, \dots, \theta^P = z^P | Y_{0:t}). \end{aligned}$$

$(\theta^1_{\tau+1}, \theta^2_\tau, \dots, \theta^P_\tau)$ is then distributed according to the law $p(\theta^1, \theta^2, \dots, \theta^P | Y_{0:t})$. In the same way we can show that $(\theta^1_{\tau+1}, \theta^2_{\tau+1}, \dots, \theta^P_\tau), \dots, (\theta^1_{\tau+1}, \dots, \theta^P_{\tau+1})$ are distributed according to $p(\theta^1, \dots, \theta^P | Y_{0:t})$.
Moreover, provided that the conditional distributions $p(\theta^1 | x, \theta^2, \dots, \theta^P), \dots,$ $p(\theta^P | x, \theta^1, \dots, \theta^{P-1})$ are strictly positive, θ_τ is irreducible. These two conditions imply the convergence for π-almost all θ_0 of $(\theta_\tau)_{\tau \in \mathbb{N}}$ towards λ.

6.2 Random number generation

We describe here the way the simulations of discrete laws are conducted. It concerns two steps of our algorithm:

- the resampling step where the discrete law is defined by the mass points $s^n_{t|t-1}$, $n = 1, \dots, N$ weighted by q^n_t , $n = 1, \dots, N$.
- the sampling of the K^j_t , $j = 1, \dots, m_t$: for j fixed, the law of K^j_t is a discrete law defined by the mass points $i = 0, \dots, M$ respectively weighted by $\pi^0_t / V, \pi^1_t l^1_t(y^j_t; x^1_t), \dots,$ $\pi^M_t l^M_t(y^j_t; x^m_t)$ (see equation (29)).

Usually, the inverse cumulative distribution function (CDF) method is used to generate random numbers according to continuous and discrete distributions. Assume a stochastic continuous variable X has an invertible CDF F_X. Then, if U is uniform over [0, 1], $X = F^{-1}_X(U)$. We have indeed

$$\mathbb{P}(F^{-1}_X(U) \leq x) = \mathbb{P}(U \leq F_X(x)) = F_X(x).$$

In the case of a discrete law X, there is a discontinuity at each mass point m_i. To simulate x according to X, we first simulate a realization u of the uniform random variable U and then pick the value x such that for $\epsilon > 0$:

$$\mathbb{P}(x - \epsilon) < u \leq \mathbb{P}(x).$$

This principle is illustrated in figure 14 below.
However it can be proved that the search of such an x implies a complexity in $O(N \log N)$ where N is the size of the sample. Because of this complexity, another algorithm is needed in the resampling step. We use the method of simulating order statistics to produce N independent ordered variables of the uniform law U ([15], [8], [2]). As these variables are already ordered the complexity of the search is reduced to $O(N)$. Let us recall how to sample such ordered realizations. Let $z_0, \dots, z_N$ be $N+1$ variables sampled from the exponential distribution. Define the variables $Z_0, \dots, Z_N$ as $Z_n = \sum_{k=0}^n z_k$. The variables $u_n = \frac{Z_n}{Z_N}$, $n = 1, \dots, N$ are then consecutive order statistics from the uniform distribution.

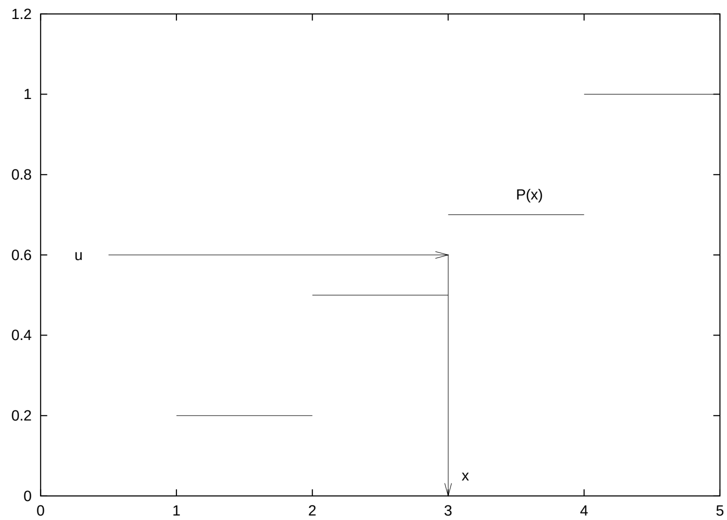


Figure 14: The inverse CDF method to deliver a number from a discrete distribution from a uniform random number

References

[1] Bar-Shalom and Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[2] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proc.-Radar. Sonar Navig*, 146(1), February 1999.

[3] G. Celeux and J. Diebolt. A stochastic approximation type em algorithm for the mixture problem. *Stochastic and Stochastics Reports*, 41:119–134, 1992.

[4] J. Diebolt and C. P. Robert. Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society series B*, 56:363–375, 1994.

[5] A. Doucet. On sequential simulation-based methods for bayesian filtering. Technical report, CUED/F-INFENG/TR 310, Signal Processing Group, Departement of Enginerring, University of Cambridge, 1998.

[6] Thomas E. Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8:173–184, July 1983.

[7] H. Gauvrit, J-P Le Cadre, and C.Jauffret. A formulation of multitarget tracking as an incomplete data problem. *IEEE. Trans. on Aerospace and Electronic Systems*, 33(4):1242–1257, October 1997.

[8] I. Geronditis and R.L. Smith. Monte carlo generation of order statistics from general distributions. *J. Royal Statist. Soc.*, 31(3), 1982.

[9] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proc.F, Radar and signal procesing*, 140(2):107–113, April 1993.

[10] M. Hürzeler and H.R. Künsch. Monte carlo approximations for general state space models. *Journal of Computational and Graphical Statistics*, 7:175–193, 1998.

[11] M. Isard and A. Blake. CONDENSATION — conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.

[12] A. Kong, J.S. Liu, and W.H Wong. Sequential imputation method and bayesian missing data problems. *J. Amer. Statist. Assoc.*, 89:278–288, 1994.

[13] X. Li and V. Jilkov. A survey of maneuvering target tracking:dynamic models. In *Conference on Signal and Data Processing of Small Targets*, Orlando, April 2000.

[14] J.S Liu. Metropolized independent sampling with comparison to rejection sampling and importance sampling. *Statistics and Computing*, 6:113–119, 1996.

[15] D. Lurie and H.O. Hartley. Machine-generation of order statistics for monte carlo computations. *The American Statistician*, 26:26–27, February 1972.

[16] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. Int. Conf. Computer Vision*, pages 572–578, 1999.

[17] C. Musso and N. Oudjane. Particle methods for multimodal filtering. In *The 2nd international Conference on information fusion IEEE*, Silicon Valley, CA, 6-8 July 1999.

[18] C. Rago, P. Willett, and R. Streit. A comparison of the JPDA and PMHT tracking algorithms. In *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pages 3571–3573, May 1995.

[19] M. Stephens. *Bayesian Methods for Mixtures of Normal Distributions.* PhD thesis, Magdalen College, Oxford, 1997.

[20] Roy L. Streit and Tod E. Luginbuhl. Maximum likelihood method for probabilistic multi-hypothesis tracking. In *Proceedings of SPIE International Symposium, Signal and Data Processing of Small Targets 1994*, volume 2335, Orlando, FL, Apr. 5-7, 1994, 1994.

[21] P. Willett, Y. Ruan, and R. Streit. The PMHT for maneuvering targets. In *Conference on Signal and Data Processing of Small Targets*, volume 3373, SPIE Annual international symposium on aerosense, Orlando, FL, April 1998.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399