Image and Vision Computing 28 (2010) 111–123

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

A Bayesian approach to joint tracking and identification of geometric shapes in video sequences

Pierre Minvielle [a,*], Arnaud Doucet [b], Alan Marrs [c], Simon Maskell [c]

[a] CEA, DAM, CESTA, F-33114 Le Barp, France
[b] The Institute of Statistical Mathematics, Tokyo, Japan
[c] QinetiQ Ltd., St. Andrews Road, Malvern, United Kingdom

ARTICLE INFO

Article history:
Received 21 November 2007
Received in revised form 24 April 2009
Accepted 2 May 2009

Keywords:
Bayesian model selection
Particle filtering
Target tracking

ABSTRACT

A Bayesian approach is proposed for joint tracking and identification. These two problems are often addressed independently in the literature, leading to suboptimal performance. In a Bayesian approach, a prior distribution is set on both the hypothesis space and the associated parameter space. Although this is straightforward from a conceptual viewpoint, it is typically impossible to perform inference in closed-form. We discuss an advanced particle filtering approach to solve this computational problem and apply this algorithm to joint tracking and identification of geometric forms in video sequences.

1. Introduction

Numerous applications in vision, defence and robotics involve tracking a variety of objects/targets of potentially different types; e.g., a human or a vehicle. Broadly speaking, the tracking task consists of determining the kinematic state of the target (e.g., its position, velocity, etc.) whereas the identification task involves characterising its identity through the estimation of attributes or features (e.g., size, shape or spectral signature) of the target.

In the past, tracking and identification have mostly been performed separately although there are a few exceptions [26,27]. While this is clearly suboptimal, it is partly due to the fact that different sensors have usually been dedicated to each task; e.g., radar for kinematic tracking, cameras for classification. The techniques employed for tracking and classification are quite different and this does not favour data fusion. Broadly speaking, tracking has been principally based on sequential estimation methods, such as Kalman filtering and all its derivatives, whilst identification techniques have been mostly feature-based [1] (i.e., using pre-established relationships between the target features and its identity), relying on pattern recognition methods such as clustering techniques, template matching or neural networks.

Although this clear division between tracking and identification still remains widespread for historical reasons, there are many applications where it is not justified anymore and joint tracking and classification could be very fruitful. This is because:

- With the development of multi-sensor data-fusion techniques, this partition between "tracking sensors" and "identification sensors" has become less and less marked. "Identification sensors", when part of a sensor network (e.g., bi-static imaging sensors), have tracking capabilities and, conversely, information produced by "tracking sensors" can be exploited to identify the target (e.g., target-specific Doppler signatures are acquired by a millimeter wave radar).
- Kinematic tracking gives information (e.g., deceleration, manoeuvrability) that characterises the behaviour of the target and may be useful to identify it. Conversely, the determination of the target identification provides information about its future dynamics and ought to improve localisation accuracy.
- In the context of multi-target tracking, tracking and identification are closely linked through the data association problem. On the one hand, improved tracking facilitates identification. On the other hand, improved identification may help to reduce the data association ambiguity.
- More importantly, tracking and identification are linked because the measurements often both depend on the kinematic state and on the identity of the target (e.g., imagery of a plane is a function of its range and orientation, but also of its type and shape).

As an alternative to feature-based identification methods, model-based methods [1,2] rely on statistical models of the targets. From the data collected by sensors, the identity of the target is

* Corresponding author. Tel.: +33 557044150; fax: +33 557045457.
E-mail addresses: pierre.minvielle@cea.fr (P. Minvielle), arnaud@ism.ac.jp (A. Doucet), admarrs@qinetiq.com (A. Marrs), s.maskell@signal.qinetiq.com (S. Maskell).

0262-8856/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.05.002

inferred. These statistical models rely on prior physical information and allow us to incorporate explicit and meaningful knowledge of the potential targets. Consequently, they sometimes allow us to bypass the training stage of supervised feature-based approaches. Moreover, such a Bayesian approach naturally also allows us to address jointly the tracking and identification problems. This overcomes the usual weakness of the featured-based identification methods; the lack of robustness to errors in the data association process. This principled methodology offers a richer way to describe the target classes, not restricted to feature values, but based on dynamic and measurement models. In this approach, classes can be represented by Bayesian models; a more general form of parametric models where fixed, but a priori unknown, parameters represent degrees of freedom, i.e., uncertainty within a class model. The parameters can also be interpreted as attributes or features of the target and can be class dependent.

The main problem associated with this Bayesian (model-based) approach is that it is impossible to perform inference analytically as it is necessary to compute high dimensional integrals. Since the beginning of the 1990s, the introduction of Markov chain Monte Carlo (MCMC) methods in statistics has truly revolutionized the practice of Bayesian statistics. Although such methods are applicable to our problem, they are iterative algorithms and are inadequate for on-line processing as required in most tracking applications. Thankfully, there is an on-line alternative to MCMC known as Sequential Monte Carlo (SMC) methods or particle filtering methods [3–5]. These techniques allow us to perform sequential Bayesian inference for any non-linear non-Gaussian state-space model [3,5]. In recent years SMC techniques have been developed to address joint tracking and identification problems [6–10] and, more generally, sequential model selection [12–15] problems. The difficult problem of estimating the fixed real-valued hyper-parameters has also been addressed [16,17]. These approaches estimate both a dynamic state and the static values of potential hyper-parameters (e.g., the target identity).

In this article, our aim is to review some of the current SMC approaches to on-line joint tracking and identification. The theoretical problems associated with these methods are discussed and we present a way to partially bypass them using an advanced SMC method. We illustrate the resulting methodology on a extended-object tracking and identification application; the application consists of 6D tracking (i.e., both the position and the orientation) and identification of basic geometric shapes moving in front of a digital CCD camera.

The rest of the paper is organised as follows. In Section 2, we present a Bayesian model for joint tracking and identification. We review standard SMC approaches to solve the associated computational problems and emphasise the limitations of standard algorithms to perform sequential inference in the presence of static hyper-parameters. In Section 3, we describe an advanced SMC approach to address the joint tracking and identification problem which overcomes some of these limitations. Finally, Section 4 presents an application of this method to geometric shapes in a video sequence. Results are given in Section 5.

2. Bayesian model and standard SMC methods

2.1. State-space models

State-space models are a popular class of statistical models which has been extensively used in tracking, signal processing, robotics and econometrics [3,4]. Formally a state-space model is defined as follows. Let $\{X_n\}_{n\geqslant 0}$ be an unobserved Markov model of initial density $p(x_0)$ and transition density $p(x_n|x_{n-1})$; that is

$$X_0 \sim p(x_0), \quad X_n|(X_{n-1}=x_{n-1}) \sim p(x_n|x_{n-1}) \tag{1}$$

We only have access to the observation process $\{Y_n\}_{n\geqslant 1}$ which is such that, conditional upon $\{X_n\}_{n\geqslant 0}$, the observations are statistically independent and distributed according to

$$Y_n|(X_n=x_n) \sim p(y_n|x_n) \tag{2}$$

For any sequence $\{z_n\}$, we denote $z_{i:j}=(z_i,z_{i+1},\ldots,z_j)$. The aim of optimal filtering is to recursively estimate $p(x_{0:n}|y_{1:n})$ or its associated marginal distribution $p(x_n|y_{1:n})$. It is straightforward to check that

$$p(x_{0:n}|y_{1:n}) = p(x_{0:n-1}|y_{1:n-1})\frac{p(y_n|x_n)p(x_n|x_{n-1})}{p(y_n|y_{1:n-1})} \tag{3}$$

where

$$\begin{aligned} p(y_n|y_{1:n-1}) &= \int p(y_n|x_n)p(x_n|x_{n-1})p(x_{n-1}|y_{1:n-1})dx_{n-1:n} \\ &= \int p(y_n|x_n)p(x_n|y_{1:n-1})dx_n \end{aligned} \tag{4}$$

Except for very simple state-space models, including linear Gaussian state-space models and hidden finite-state space Markov chains, it is impossible to evaluate these expressions analytically. SMC methods are a set of numerical methods where the distributions of interest are approximated by a large number N of random samples termed particles which are propagated forward in time using sequential importance sampling and resampling steps. We refer the reader to [3–5] for a more complete description of SMC algorithms.

Assume we have N samples or particles $X_{0:n}^{(i)}$ of respective weights $W_n^{(i)}$ $(i=1,\ldots,N)$, which provide the following SMC approximation of $p(x_{0:n-1}|y_{1:n-1})$ at time $n-1$:

$$\widehat{p}(x_{0:n-1}|y_{1:n-1}) = \sum_{i=1}^{N} W_{n-1}^{(i)}\delta_{X_{0:n-1}^{(i)}}(x_{0:n-1}) \tag{5}$$

where $\delta_{x_0}(x)$ denotes the Dirac delta mass located at x_0.

To obtain an approximation $\{W_n^{(i)},X_{0:n}^{(i)}\}$ of $p(x_{0:n}|y_{1:n})$ at time n, the SMC algorithm proceeds as follows:

- For $i=1,\ldots,N$, draw $X_n^{(i)} \sim q(x_n|y_n,X_{n-1}^{(i)})$, where $q(x_n|y_n,x_{n-1})$ is an importance density, which is a probability density used to sample $X_n^{(i)}$ given $(y_n,X_{n-1}^{(i)})$.
- For $i=1,\ldots,N$, compute

$$W_n^{(i)} \propto W_{n-1}^{(i)}\frac{p\left(y_n|X_n^{(i)}\right)p\left(X_n^{(i)}|X_{n-1}^{(i)}\right)}{q\left(X_n^{(i)}|y_n,X_{n-1}^{(i)}\right)}, \quad \text{with } \sum_{i=1}^{N} W_n^{(i)}=1 \tag{6}$$

Straightforward implementation of the first two steps will typical result in a situation where after a few iterations, all but one particle will have negligible weight. This is known as the so-called degeneracy problem and the approximation accuracy is strongly degraded. Following [3], it is possible to compute a suitable measure of degeneracy $N_{eff}=(\sum_{i=1}^{N}W_n^{(i)^2})^{-1}$, called the effective sample size. If it falls below a threshold N_T (typically pre-set to $N/2$), resample the particles from:

$$\widehat{p}(x_{0:n}|y_{1:n}) = \sum_{i=1}^{N} W_n^{(i)}\delta_{X_{0:n}^{(i)}}(x_{0:n}) \tag{7}$$

and reset the weights to $W_n^{(i)}=1/N$. The samples are then also denoted abusively $\{X_{0:n}^{(i)}\}$.

This algorithm provides an approximation $\hat{p}(x_{0:n}|y_{1:n})$ of the posterior at time n and also an estimate of the so-called evidence or marginal likelihood:

$$\widehat{p}(y_{1:n}) = \prod_{k=1}^{n}\widehat{p}(y_k|y_{1:k-1}) \tag{8}$$

where $y_{1:0} = \emptyset$ and

$$\hat{p}(y_k|y_{1:k-1}) = \sum_{i=1}^{N} W_{k-1}^{(i)} \frac{p\left(y_k|X_k^{(i)}\right) p\left(X_k^{(i)}|X_{k-1}^{(i)}\right)}{q\left(X_k^{(i)}|y_k, X_{k-1}^{(i)}\right)} \tag{9}$$

2.2. State-space models with model uncertainty

In the context of joint tracking and identification, the state-space model takes the following form. First of all, we introduce the following assumptions:

- There are M possible models for the target identity arbitrary labelled $\{1,\ldots,M\}$ and the prior distribution is given by

  $\Pr(I = k) = p_k$

- Each model k has potentially unknown hyper-parameters $\theta \in \Theta_k$ of prior distribution $p_k(\theta)$.
- The random vector (I,θ) takes values in $\cup_{k=1}^{M}\{k\} \times \Theta_k$ which is a union of subspaces of potentially different dimension.

Conditional on $(I,\theta) = (k,\theta)$, the target follows a standard state-space model

$$\begin{aligned} &X_0 \sim p(x_0|k,\theta), \quad X_n|(X_{n-1} = x_{n-1}) \sim p(x_n|x_{n-1},k,\theta) \\ &Y_n|(X_n = x_n) \sim p(y_n|x_n,k,\theta) \end{aligned} \tag{10}$$

Inference is based on the joint posterior $p(i,\theta,x_{0:n}|y_{1:n})$ at time n which follows the recursion

$$p(i,\theta,x_{0:n}|y_{1:n}) = p(i,\theta,x_{0:n-1}|y_{1:n-1}) \frac{p(y_n|x_n,k,\theta)p(x_n|x_{n-1},k,\theta)}{p(y_n|y_{1:n-1})} \tag{11}$$

where

$$\begin{aligned} p(y_n|y_{1:n-1}) &= \sum_{i=1}^{M} \int_{\Theta_i} \int p(y_n|x_n,i,\theta)p(x_n|x_{n-1},i,\theta)p(i,\theta,x_{n-1}|y_{1:n-1})d\theta dx_{n-1:n} \\ &= \sum_{i=1}^{M} \int_{\Theta_i} \int p(y_n|x_n,i,\theta)p(i,\theta,x_n|y_{1:n-1})d\theta dx_n \end{aligned}$$

The posterior probabilities of the models are given by

$$p(i|y_{1:n}) = \frac{p(y_{1:n}|i)p_i}{\sum_{k=1}^{M} p(y_{1:n}|k)p_k} \tag{12}$$

Once more, it is impossible to evaluate these posterior distributions analytically.

2.3. Limitations of standard SMC algorithms

Formally SMC can directly be applied to approximate $p(i,\theta,x_{0:n}|y_{1:n})$ and $p(i|y_{1:n})$. Indeed, it only requires modifying the initialization step of the algorithm described in Section 2.1 to sample additionally N particles $\{I^{(i)},\theta^{(i)}\}$ according to their prior distribution. Subsequently, we propagate the particles $\{I^{(i)},\theta^{(i)},X_{0:n}^{(i)}\}$ instead of $\{X_{0:n}^{(i)}\}$ in the standard approach. Unfortunately, although it can be proved that as the number of particles $N \to \infty$ the resulting SMC approximation converges towards $p(i,\theta,x_{0:n}|y_{1:n})$ under weak assumptions [3,28], this convergence is not uniform in time. This means that, as the time index increases, we would need an increasing number of particles to ensure a fixed precision of the approximation. The reason for this problem is that static parameters are sampled at time $n = 0$ but they are subsequently never rejuvenated; this is the so-called degeneracy problem. Particles are only depleted by the resampling step; this results in a rapid collapse of the approximation.

3. SMC methods for joint tracking and identification

In this section, we describe the main approaches to joint tracking and identification, separated according to our closely linked goals: the *intra-class goal* (i.e., joint tracking and estimation of the target attributes, or parameter estimation for a given class) and the *inter-class goal* (i.e., the classification of the target or model selection).

3.1. Continuous target attribute estimation

3.1.1. Artificial evolution methods

An immediate and pragmatic way to avoid the degeneracy problem is to adopt an approach that has been applied previously in the Kalman filtering framework, i.e., to introduce explicitly an artificial dynamic on the hyper-parameter of interest θ

$$\theta_n = \theta_{n-1} + V_n, \quad \text{where } V_n \sim \mathcal{N}(0,\Sigma)$$
$$\text{and} \quad \Sigma \text{ is the perturbation variance}$$

The small random perturbations on the parameter reduce the memory of the system since new parameter values are generated at each time step. However, a loss of information inevitably comes from this artificial evolution since the parameter is in fact fixed. The resulting posterior distribution is more diffuse than the true posterior distribution.

Kernel density methods, used by regularised particle filters (RPF) to introduce diversity among the particles [3,5] introduce a more implicit evolution of the parameter by using a kernel density estimate of the marginal distribution of θ

$$\widehat{p}(\theta|y_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} K_h(\theta - \theta^{(i)})$$

where K_h is the regularization kernel, h the bandwidth and W_n the sample weights at the current step n.

The evolution is here adaptive and depends on the current samples. However, standard kernel methods have a tendency to overdisperse the samples and the consequent loss of information will accumulate at each time step. A variant called kernel smoothing with shrinkage [16] has been proposed to reduce this loss of information. It consists of methods that attempt to maintain the correct variance of the samples by shrinking the samples towards their mean before applying a random perturbation; see [16], for details. This method is inappropriate for multi-modal posteriors (since it involves an implicit assumption of unimodality).

All these artificial evolution methods have a tendency to artificially downweight the importance of early observations compared to recent observations. In other words, although the system is now allowed to forget its past, it also might forget informative measurements of the past, which could have been useful in determining the target attributes and eventually its identity. From a theoretical perspective, the samples are no longer guaranteed to approximate those from the true posterior even as $N \to \infty$ [5].

3.1.2. Markov chain Monte Carlo moves

The incorporation of an MCMC step within an SMC algorithm has been developed in recent years to mitigate sample impoverishment [12,13]. MCMC, previously mentioned for off-line estimation in Section 1, is a powerful technique for sampling from distributions only up to a normalising constant. MCMC can be interpreted as a principled way to "jitter" the particle locations and thus to reduce degeneracy. Let $\Lambda = (\theta, x_{0:n})$ and $\Lambda^* = (\theta^*, x_{0:n}^*)$ then an MCMC kernel $K_n(\Lambda^*|\Lambda)$ of invariant distribution $p(\Lambda|y_{1:n})$ is a Markov transition kernel with the property that

$$\int p(\Lambda|y_{1:n}) K_n(\Lambda^*|\Lambda) d\Lambda = p(\Lambda^*|y_{1:n})$$

For such a kernel, if $\Lambda \sim p(\Lambda|y_{1:n})$ and $\Lambda^*|\Lambda \sim K_n(\Lambda^*|\Lambda)$ then Λ^* is still marginally distributed according to $p(\Lambda|y_{1:n})$. Even if Λ is not distributed according to $p(\Lambda|y_{1:n})$ then, after an application of the MCMC kernel, Λ^* can only have a distribution closer than that of Λ (in total variation norm) to $p(\Lambda|y_{1:n})$. A Markov kernel is said to be ergodic if iterative application of that kernel generates samples whose distribution converges towards $p(\Lambda|y_{1:n})$ irrespective of the distribution of the initial state. Within an SMC algorithm there is no need to use many MCMC steps as we already have samples approximately drawn from the posterior distribution. Moreover, the kernel does not even need to be ergodic [13]. A generic way to build an MCMC kernel is the Metropolis–Hastings (MH) algorithm. It requires selecting a proposal distribution $q(\Lambda^*|\Lambda)$. We sample a candidate according to $q(\Lambda^*|\Lambda)$ and accept it with the following probability: $\min\left\{1, \frac{p(\Lambda^*|y_{1:n})q(\Lambda|\Lambda^*)}{p(\Lambda|y_{1:n})q(\Lambda^*|\Lambda)}\right\}$.

At first sight, it appears that the implementation of MCMC steps within the particle filter requires the storage of the whole trajectory of the particles. If this were the case, both the memory requirements and the move computational cost would increase over time. However, there are many scenario where $p(\theta|y_{1:n}, x_{0:n})$ only depends on $(y_{1:n}, x_{0:n})$ through a set of fixed dimensional sufficient statistics [12,14].

Although the use of MCMC moves tackles sample impoverishment in a more rigorous way than the artificial evolution methods, a drift of the estimated parameters has been observed in [12] for long observations and a uniform (in time) convergence result cannot be established for such algorithms. Storvik [17] proposes a method based on marginalization of the parameter from the posterior. Note that this method also suffers from the degeneracy problem.

3.2. Identity determination

3.2.1. Class-specific filters

Class-specific filters have been developed [6] for joint target tracking and feature estimation problems where the target feature is discrete and can only take a finite number of values. This problem description applies to the identification determination when the target feature refers to the identity (ID) of the target. Essentially the algorithm simply consists of running M independent particle filters to compute $p(y_{1:n}|i)$ using the fact that we can approximate the evidence using (8). The method benefits from the system conditional ergodicity given a class. Notice that computation load remains constant for each class, even if the class becomes very unlikely. However, this extra load makes the method robust and avoids misclassifications by enabling the filter to "change its mind". Refs. [6,7] illustrate the efficiency of the approach for behaviour-based classification problems.

3.2.2. Sequential Reversible Jump MCMC

Reversible Jump MCMC is a general methodology, based on the Metropolis–Hastings (MH) algorithm, which addresses the problem of off-line model selection [11,23]. Its purpose is to construct an ergodic Markov chain admitting the global joint posterior distribution (here $p(i, \theta, x_{0:k}|y_{1:k})$) as the invariant distribution. Compared to basic MCMC, Reversible Jump MCMC tackles the difficult task of jumping from one model to another model. Applied to sequential identification with SMC, Reversible Jump MCMC extends the capabilities of the previously mentioned MCMC step or Resample-Move Filter. Reversible Jump MCMC algorithms can consist of a mixture of kernels generating intra-classes moves inside subspaces and extra-classes cross-model moves. Furthermore, birth and death moves can be proposed to explore new or depleted areas of the space. Efficiently designed, all those moves can reduce efficiently the impact of sample impoverishment and redistribute the particles across the whole joint posterior $p(i, \theta, x_{0:n}|y_{1:n})$. All the previous remarks for the MCMC moves are still valid.

Note that the class probabilities are here simply determined by the proportion of samples associated with the various classes. Unlike the class-specific filters approach, the same computational load is not attributed to each class; classes turning out to be very unlikely will require very small efforts. Moreover, cross-model jumps are supposed to be able to repopulate occasionally depleted class subspaces. However, engineering efficient reversible moves is known to be a very tricky, time-consuming task [23]. Moreover the Reversible Jump MCMC approach is probably more useful when the number of potential classes is quite high and when it is possible to take advantage of relationships between parameters of different classes to propose efficient cross-model moves.

3.3. Comments

None of the SMC approaches for joint tracking and identification presented here significantly outperforms the others. The choice of an appropriate SMC strategy is problem-dependent. Class-specific filters prove to be especially efficient and robust when the various expected target classes can be modelled in very different ways and can require completely different inference processes. On the other hand, the Reversible Jump MCMC methodology is more appropriate to deal with a large (possibly infinite) number of classes. Focusing on fixed-parameter estimation, marginalization and sufficient statistics should be used whenever possible. Furthermore, disengagement procedures [17,22] may be applicable in some contexts. These procedures consist of neglecting some earlier data points. For example, disengagement can take the form of a sliding window (e.g., [19] in the context of communication processing) where only recent measurements are considered, or the form of a deliberate concealment of previous measurements that a specific event could validate (e.g., [22] in the context of clinical monitoring). In less demanding applications, another solution can be to rejuvenate the particle approximation from time to time by the use of full off-line MCMC algorithms [15]. As a last resort, an artificial dynamic model may be introduced on the static parameter.

The listed approaches are not necessarily used in isolation of one another. Indeed, the combination of various techniques may be an efficient and useful way to address the joint tracking and identification task under consideration. The following video tracking application illustrates such an hybrid approach.

4. Application to geometric shape video tracking

In this section, we describe an application of the methodology presented earlier to digital CCD video tracking of basic geometric shapes (such as cubes, parallelepipeds, spheres and hemispheres). The purpose of the application is not to deal with a real computer vision problem but to illustrate how SMC techniques can address joint tracking and identification problems. Moreover, the application demonstrates how different SMC techniques can be combined efficiently.

In the tracking scenario, a shape moves inside the field of view (FOV) of a CCD camera, alternatively getting closer and moving away while rotating (cf. Fig. 1). The application goals are:

- *3D tracking:* to track the shapes in the real 3D space, i.e., estimate simultaneously the position and the orientation of the moving shape (the so-called target pose),
- *attribute estimation:* to estimate sequentially also the dimensions of the shape if they are unknown (e.g., radius of a sphere, length/width/height of a parallelepiped),

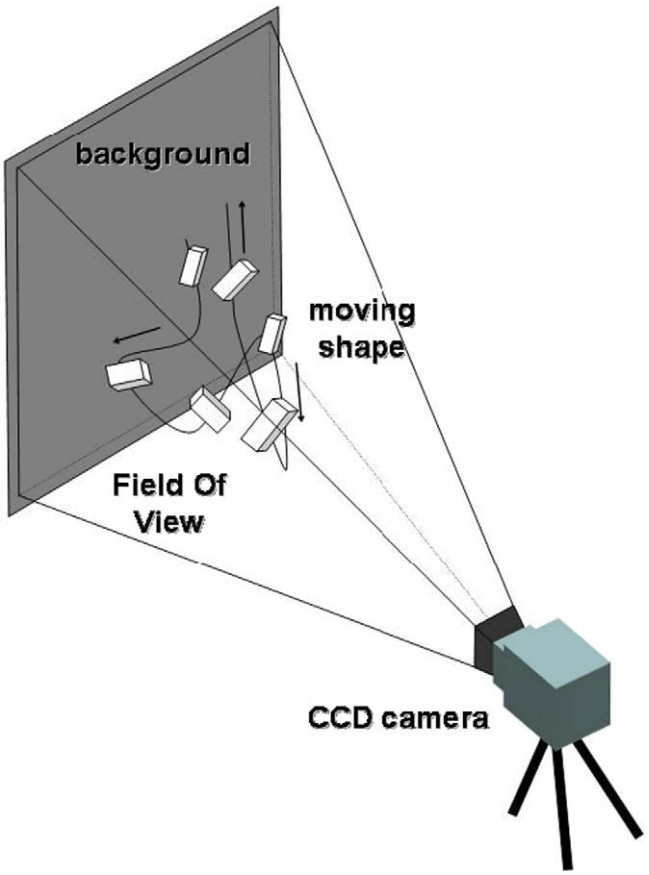

Fig. 1. The tracking scenario.

– *identification:* to identify the shape, when there is uncertainty about the class (e.g., whether the shape is a sphere or hemisphere, cube or parallelepiped).

Note that in order to focus on the tracking issues and not to deal with very specific image processing problems, the scenario has been simplified. The environment is uncluttered (a black cloth has been laid behind the target). The lighting conditions are satisfactory, providing good conditions for feature extraction. Moreover, there is no occlusion of the target (the target is suspended by threads).

The main characteristics (cf. Fig. 2) of the developed Bayesian approach are:

- *The parallel architecture:* It is based on *class-specific filters* [6]. Moreover, these filters are combined so as to estimate the class probabilities. With this architecture, each shape model class is able to have its own state space (its dimension differs, depending on the symmetry level of the shapes). While [6] illustrates behaviour-based identification, this application illustrates feature-based identification.
- *The inference engine:* Each class-specific filter is based on a recent particle filtering technique [18,19] which is essentially a sequential adaptation of annealed importance sampling [20]. For convenience, it will be called subsequently *simulated annealed particle filter* (SAPF). Briefly speaking, SAPF mitigates the degeneracy phenomenon by computing gradually the posterior distribution and using MCMC moves. A related technique has already been used for human body video tracking [21].
- *The hyper-parameter estimation:* The fixed parameters, which are here the dimensions of the shapes, are added to the dynamic state vector, leading to a hybrid state vector. The dimensions are estimated by combining two fixed-parameter estimation techniques. The first one consists of MCMC moves that re-scale both the trajectory and the shape; this move benefits from sufficient statistics that summarise the trajectory. The second one is an artificial evolution of deformation of the shape that diffuses across the layers of the SAPF.
- *The likelihood model:* The likelihood function uses extracted features (foreground and edges) that are provided by an image processing stage and compares them to an expected projection of the shape requiring a shape model and a camera model.

In the next sections, we detail the video application, such as the image processing, the dynamic model, etc. We describe the SAPF particle filter and focus on combined techniques to address the parameter estimation problem. Finally, the algorithm efficiency is demonstrated experimentally.

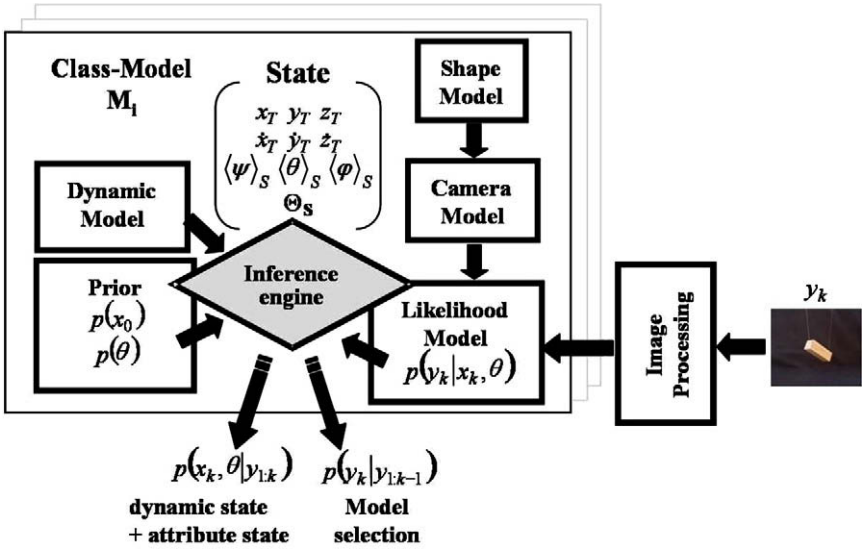

Fig. 2. Shape video tracking application.

4.1. Hybrid state-space model

In this section, we describe the state-space model which depends on the shape class. Next, the dynamic model and the prior distribution are presented.

4.1.1. Shape class-dependent state-space model

A hybrid state vector $\chi_n = [\mathbf{x}_n \theta_S]^{\mathrm{T}}$ describes the state of the target at time n. Omitting the time index n, the dynamic part is

$$\mathbf{x} = [x_T\ y_T\ z_T\ \dot{x}_T\ \dot{y}_T\ \dot{z}_T \langle\ \Psi\rangle_S \langle\zeta\rangle_S\ \langle\varphi\rangle_S]$$

where

- $[x_T y_T z_T]$ and $[\dot{x}_T \dot{y}_T \dot{z}_T]$ are, respectively, the location and the velocity vector of the target (more exactly of the barycentre T) in the camera coordinate system (OXYZ) (cf. Fig. 3),
- $[\langle\Psi\rangle_S \langle\zeta\rangle_S \langle\varphi\rangle_S]^{\mathrm{T}}$ defines the orientation of the target. More precisely, they are Euler angles (Ψ, ζ, φ) which define a sequence of three standard rotations that transforms the target coordinate system $(T_x T_y T_z)$ into the general coordinate system (XYZ).

The brackets $\langle\rangle_S$ means that some of the Euler angles may be optional, depending on the level of symmetry of the shape. There is no orientation for spheres and the associated state vector does not include any Euler angles. For hemispheres, their orientation is given by their axis of symmetry and (Ψ, φ) is enough. For cubes or parallelepipeds, all the angles (Ψ, ζ, φ) are required (see Fig. 4).

The parameter part θ_S depends on the shape: $\theta_S = r$ (radius) for spheres and hemispheres, $\theta_S = l$ (length) for cubes, $\theta_S = [lwh]^{\mathrm{T}}$ (length, width, height) for parallelepipeds.

To summarise, each shape model class has its own dedicated state space, the dimension of which is varying: 7 for the class sphere, 9 for the class hemisphere, 10 for the class cube and 12 for the class parallelepiped.

4.1.2. Dynamic model

The dynamic model is a (second order) piecewise constant white acceleration model [24] on the position component:

$$x_n = x_{n-1} + \Delta T \dot{x}_{n-1} + \frac{1}{2}\Delta T^2 v_n^x, \quad \dot{x}_n = \dot{x}_{n-1} + \Delta T v_n^x \tag{13}$$

where v_n^x is a white Gaussian noise of variance σ_v^2. Identical models are adopted for y_n and z_n. As for the orientation component, the related dynamic model is a random walk

$$\Psi_n = \Psi_{n-1} + \varepsilon_n^{\Psi}, \quad \zeta_n = \zeta_{n-1} + \varepsilon_n^{\zeta}, \quad \varphi_n = \varphi_{n-1} + \varepsilon_n^{\varphi} \tag{14}$$

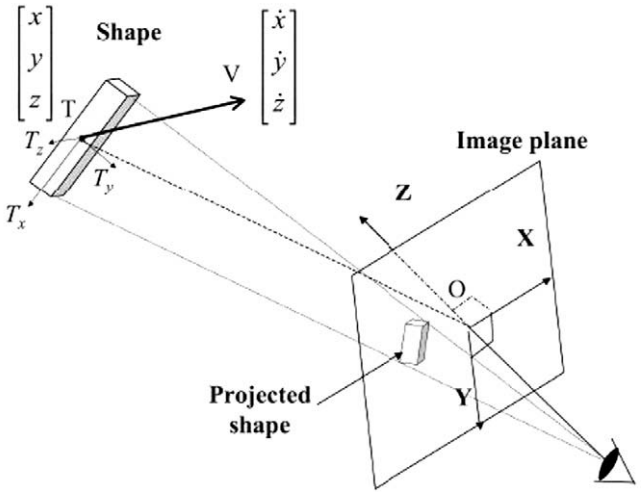

Fig. 3. The pinhole camera model.

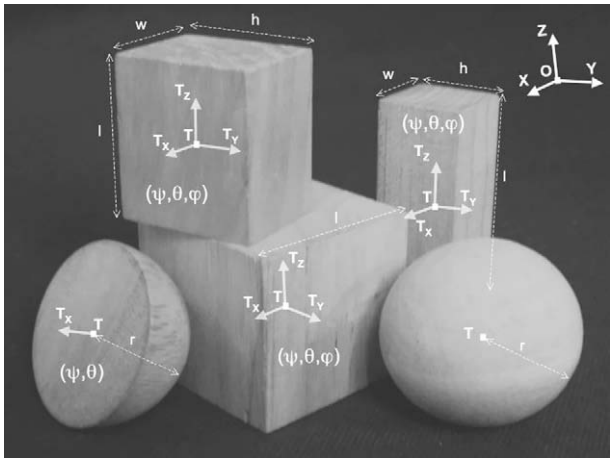

Fig. 4. State space and symmetry level.

where $\varepsilon_n^{\Psi}, \varepsilon_n^{\theta}, \varepsilon_n^{\varphi}$ are white Gaussian noise independent sequences, mutually independent, of variance σ_{ε}^2.

4.1.3. Prior distribution

The prior distribution $p(\mathbf{x}_0 \theta_S) = p(\mathbf{x}_0) p(\theta_S)$ is simply given by:

- *the state prior* $p(\mathbf{x}_0)$*:* it is simply described by uniform distributions on the range, on the position of the target in the image plane $p(x_0, y_0)$, on its orientation and velocity,
- *the parameter prior* $p(\theta_S)$*:* similarly to the state prior, it is described by an uniform distribution on each parameter of the target.

4.2. Observation model

4.2.1. Image processing

Similarly to [21], a basic pre-processing feature extraction step is applied to the current image. Two features are extracted: the edges and the foreground (cf. Fig. 5). The foreground extraction consists of a straightforward thresholding of the image intensity, followed by morphological operations to reject detections associated with the threads and resulting in a binary pixel map I_F. On the other hand, a gradient-based Canny detector is used to extract the edges; the result, smoothed with a Gaussian 2D filter, is remapped between 0 and 1. Eventually, the pixels of the resulting pixel map I_E are assigned a value related to their proximity to an edge. Notice that both the lighting effects (e.g., when surfaces are nearly at right angles to the image plane) and the blurring effects may cause some of the edges to be undetected (cf. right part of Fig. 5).

4.2.2. Shape and camera models

Associated with a shape particle and its state vector $\chi_n = [\mathbf{x}_n \theta_S]^{\mathrm{T}}$ at time n, a shape model and a camera model are used to determine the expected image features of such a particle. The shape model generates edge and foreground sampling points, according to the class of the shape and its pose (i.e., its position $[x_T y_T z_T]$ and its orientation $[\langle\Psi\rangle_S \langle\zeta\rangle_S \langle\varphi\rangle_S]$). The N_E sampling points on the expected visible edges $\{E_k^{x,\theta}\}_{k=1}^{N_E}$ (black circles in Fig. 6) are regularly drawn while the N_F sampling points on the expected foreground $\{F_k^{x,\theta}\}_{k=1}^{N_F}$ (gray squares) are randomly spaced. Then, the locations of the sample points in the image plane are determined by a pinhole camera model (see Fig. 3). Notice that, though simple, such a model is non-linear.

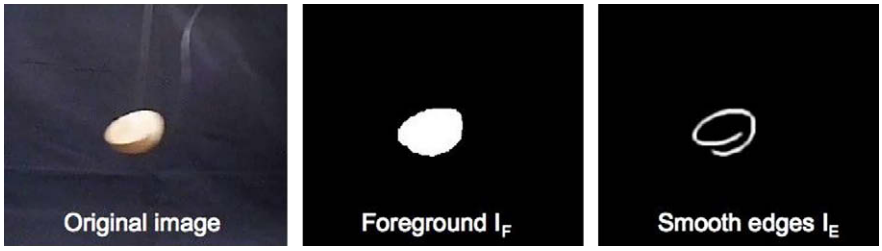

Fig. 5. Feature extraction.

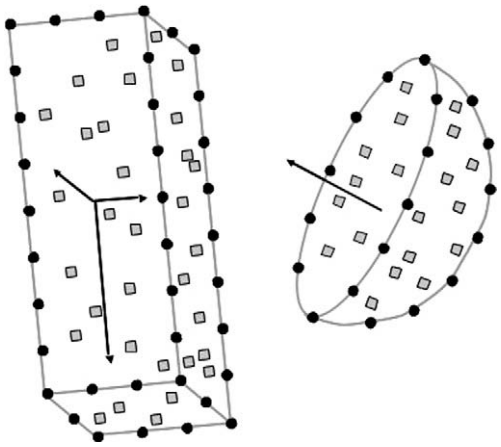

Fig. 6. Feature sampling points.

4.3. Likelihood function

The likelihood model strongly depends on the application and on the chosen image features. In a similar way to [21], the likelihood function $p(y|x,\theta)$ is constructed as the product of the likelihood $L_y^E(x,\theta)$ and $L_y^F(x,\theta)$ based, respectively, on the edges and the foreground:

- L_y^E relies on the following metric: $L_y^E(x,\theta) = T_y^M(x,\theta)\cdot T_y^U(x,\theta)$.

T_y^M corresponds to a distance between matched edges of the model and the image. It is determined by the following expression

$$T_y^M(x,\theta) = \prod_{k=1}^{N_E}(A + B\cdot \mathscr{I}_{I_E}(E_k^{x,\theta}))$$

where $\mathscr{I}_{I_E}$ is the deduced (bilinear interpolation) intensity on the normalised pixel map I_E of the expected visible edge sampling point $E_k^{x,\theta}A$ and B are, respectively, tuned to 0.8 and 0.2 so that T_y^M cannot exceed 1. On the other hand, T_y^U introduces a penalty for the unmatched edges:

$$T_y^U(x,\theta) = \prod_{\text{pixel } p|d(p,E_k^{x,\theta})\geqslant d_V \forall k=1,\ldots,N_E} (1 - C\cdot I_E(p))$$

The more image pixels with significant intensity outside the neighborhood (delimited by the pixel distance d_V) of the model sampling point $E_k^{x,\theta}$, the lower T_y^U is. $C = 0.1$ and $d_V = 5$ for image resolution 320×240.

- $L_y^F(x,\theta)$ is linked to the rate of samples from $\{F_k^{x,\theta}\}_{k=1}^{N_F}$ lying in the extracted foreground map I_F:

$$L_y^F(x,\theta) = \prod_{k=1}^{N_F}\alpha^{1-I_F(F_k^{x,\theta})} = \alpha^{N_F\cdot\left(1-\frac{K_F}{N_F}\right)}$$

K_F is the evaluated number of the foreground model samples inside the extracted foreground. α is set to 0.9. The higher K_F is, the higher the foreground likelihood is.

Notice that the various constants are tuned so that L_y^E and L_y^F, as well as the matched and unmatched edges components, are appropriately weighted.

4.4. Simulated annealed particle filter

In this section, we present the generic SAPF filter and the specific implementation designed for the video application.

4.4.1. Principles

The SAPF filter [20] can be also interpreted as a Resample-Move algorithm [14]; see [25] for a discussion and a generalisation of such techniques. SAPF is useful when no good importance density is available, meaning when the likelihood is centred far away from the points sampled use the importance density. For example, this is the case when, through lack of an efficient importance density, the prior is used despite being much more diffuse than the likelihood. The basic idea of the SAPF is to use a sequence of intermediate distributions to move smoothly towards the target distribution.

Between the initial distribution $\Pi_0(x_{0:n}) = p(x_{0:n-1}|y_{1:n-1})q(x_n|x_{n-1},y_{1:n})$ (q is the importance density) and the final distribution $\Pi_0(x_{0:n}) = p(x_{0:n}|y_{1:n}), P-1$ intermediate distributions $\Pi_m(x_{0:n})$ are defined according to an appropriate schedule, such as:

$$\Pi_m(x_{0:n}) \propto p(x_{0:n-1}|y_{1:n-1})q(x_n|x_{n-1},y_n)^{\alpha_m}p(y_n|x_n)^{1-\alpha_m} \tag{15}$$

The sequence $\{\alpha_m\}$ is the cooling schedule and satisfies $\alpha_0 = 1 < \alpha_1 < \cdots < \alpha_M = 0$. The SAPF aims to move the set of particles through this sequence of distributions.

Given an approximation $\{X_{0:n-1}^{(i)}, W_{n-1}^{(i)}\}$ of $p(x_{0:n-1}|y_{1:n-1})$ at time $n-1$, we obtain an new approximation $\{X_{0:n}^{(i)}, W_n^{(i)}\}$ of $p(x_{0:n}|y_{1:n})$ at time n as follows for the sequence of intermediate distributions (15).

- For $i = 1,\ldots,N$, draw $X_n^{(i),(0)} \sim q\left(x_n|y_n, X_{n-1}^{(i)}\right)$.
- For $m = 1,\ldots,P+1$
  - Set $X_n^{(i),(m)} = X_n^{(i),(m-1)}$ and compute $W_n^{(i),(m)} \propto W_n^{(i),(m-1)}p(y_n|X_n^{(i),(m-1)})^{\alpha_m-\alpha_{m-1}}$.
  - If $\left(\sum_{i=1}^N W_n^{(i),(m)}\right)^{-1} < N_T$ then resample N particles also denoted $\{X_{0:n}^{(i),(m)}\}$ from the weighted distribution $\{X_{0:n}^{(i),(m)}, W_n^{(i),(m)}\}$ and set $W_n^{(i),(m)} = 1/N$.
  - Apply an MCMC move of invariant distribution $\Pi_m(x_{0:n})$.

The algorithm is illustrated in Fig. 7 with a number of simulated annealing layers equal to 3. Starting from the importance distribution Π_0 (black), the likelihood is progressively introduced, so that at each layer the particles concentrate themselves progressively in the probable parts of the state space. Eventually, the particles are (approximately) distributed according to the posterior distribution Π_3.

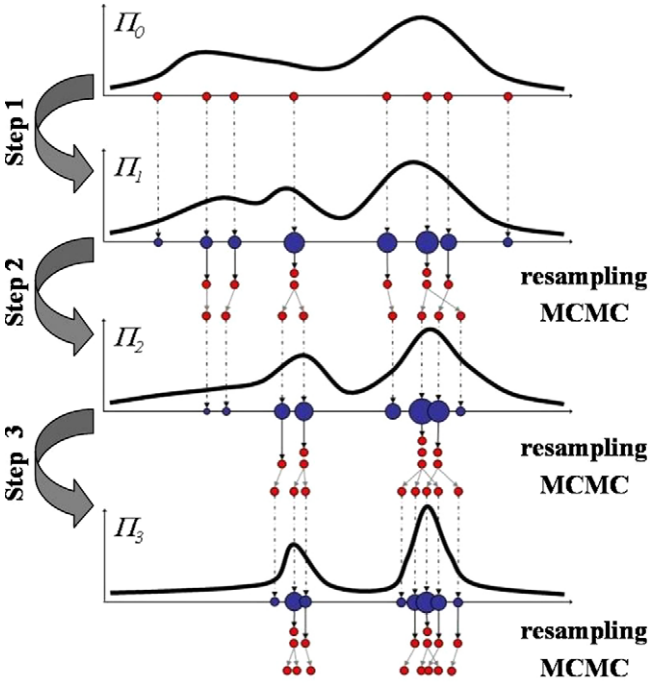

Fig. 7. The SAPF principle.

Despite the lack of a good importance distribution, the SAPF algorithm is able to deal with a multi-modal likelihood without getting trapped in local maxima. An appropriate cooling schedule [19] is implemented: as the dimension of the state space becomes higher and the search of the good modes more challenging, it monitors the size of the successive steps. It is also straightforward and natural to add target attribute moves inside the MCMC move step in order to perform fixed-parameter estimation. Concerning the model selection goal, the class-specific filters approach described in Section 3.2.1 is chosen.

The main MCMC move is a kinematic move where the position, the velocity and the orientation of the target are simultaneously updated. The variance of the proposal distribution decreases at each layer of the SAPF recursion as large moves become less and less possible. The variance evolution is tuned in order to ensure that around 30% of the moves are accepted. The other MCMC moves, dedicated to the target features, are presented further.

4.5. Combined techniques for shape feature estimation

Combined techniques are used to estimate the shape features. The first technique is a resample-MCMC move that is here applied to re-scale both the dimensions of the target and the whole trajectory. Then, as described for the point 2D bearings-only tracking problem [14], a sufficient statistic summarises the whole trajectory so the computational and memory requirements of the re-scaling moves are reasonable. In order to be able to reach any point of the hyper-parameter space, other moves are needed. They consist in deformation moves that change the shape dimensions without keeping the proportions. Since no sufficient statistic is available, an artificial evolution is used consisting of a deformation diffusion through the layers of the SAPF.

4.5.1. Re-scaling MCMC moves

Denoting the whole trajectory Λ_k (states at all time points and parameters) at time k, the re-scaling move K_λ consists of

$$\Lambda_k = [\mathbf{x}_0, \ldots, \mathbf{x}_k \boldsymbol{\theta}_S] \overset{K_\lambda}{\to} \Lambda_k^* = [\mathbf{x}_0^*, \ldots, \mathbf{x}_k^* \boldsymbol{\theta}_S^*]$$

where $\mathbf{x}_0^*, \ldots, \mathbf{x}_k^*$ correspond to a reduction or amplification of the position and velocity parts of the states $\mathbf{x}_0, \ldots, \mathbf{x}_k$ by the same factor λ, and the orientation part remaining unchanged. As for the parameters, $\boldsymbol{\theta}_S^* = \lambda \cdot \boldsymbol{\theta}_S$, i.e., the moved shape is the homothetic of the original shape by the factor λ (the proportions remain unchanged). As a matter of fact, the move is nothing else than envisaging that the shape particle is larger or smaller, but, respectively, further or closer in such a way that the expected measurements would be unchanged.

In order to construct a Markov chain transition kernel K_λ that keeps the joint posterior distribution $p(\mathbf{x}_{0:k}, \boldsymbol{\theta}_S | y_{1:k})$ invariant, a Metropolis–Hastings method is chosen with a proposal $q(\Lambda^*|\Lambda)$. The acceptance probability of the move is:

$$\min\left\{1, \frac{p(\Lambda^*|y_{1:k}) q(\Lambda|\Lambda^*)}{p(\Lambda|y_{1:k}) q(\Lambda^*|\Lambda)}\right\} \tag{16}$$

Using the Markovian property of the dynamic process and the conditional independence of the observations given the process, the ratio $\rho(\Lambda, \Lambda^*) = \frac{p(\Lambda^*|y_{1:k})}{p(\Lambda|y_{1:k})}$ simplifies

$$\rho(\Lambda, \Lambda^*) = \underbrace{\frac{p(r_0^*) \cdot p(v_0^*) \cdot p(\boldsymbol{\theta}_S^*)}{p(r_0) \cdot p(v_0) \cdot p(\boldsymbol{\theta}_S)}}_{prior} \cdot \underbrace{e^{\frac{k\bar{\gamma}_k \cdot (1-\lambda^2)}{2\sigma_v^2}}}_{trajectory} \tag{17}$$

where $\bar{\gamma}_k = \frac{1}{k}\sum_{i=1}^k \gamma_i$ with

$$\gamma_i = \left(\frac{\dot{x}_{i+1} - \dot{x}_i}{dT}\right)^2 + \left(\frac{\dot{y}_{i+1} - \dot{y}_i}{dT}\right)^2 + \left(\frac{\dot{z}_{i+1} - \dot{z}_i}{dT}\right)^2$$

as the dynamic model satisfies $p(\mathbf{x}_{i+1}|\mathbf{x}_i) = \frac{1}{(\sqrt{2\pi}\sigma_v)^3} e^{-\frac{\gamma_i}{2\sigma_v^2}}$.

Note that if $\bar{\gamma}_k = 0$, there is no apparent motion and (17) indicates that all the re-scaling moves are possible, which is exactly the uncertainty of the information provided by only one angle measurement. In the other hand, while k increases, possible moves become smaller and smaller. The sufficient statistics $(r_0, v_0, \boldsymbol{\theta}_S, \bar{\gamma}_k)$ sum up all the information needed to evaluate the re-scaling move K_λ. Thus, it only requires adding $\bar{\gamma}_k$ to the state and to update it recursively at time k. The associated re-scaling move is then: $\bar{\gamma}_k^* = \lambda^2 \cdot \bar{\gamma}$.

As in [13], an uniform symmetric proposal density $q(\Lambda^*|\Lambda) = q(\Lambda|\Lambda^*)$ is chosen for the re-scaling move K_λ, i.e., $\lambda \sim U[\lambda_1, 1/\lambda_1]$; the value of the chosen constant λ_1 is tuned and decreases in time in order to assure more or less a rate of acceptance move around 30%. Eventually, the re-scaling move K_λ is added to the transition MCMC move K_T of the SAPF algorithm.

4.5.2. Deformation artificial diffusion

To complement the re-scaling move K_λ and to be able to reach any point of the parameter space, a complementary deformation move K_D has to be performed, consisting of:

$$\Lambda_k = [\mathbf{x}_0, \ldots, \mathbf{x}_k \boldsymbol{\theta}_S] \overset{K_D}{\to} \Lambda_k^* = [\mathbf{x}_0, \ldots, \mathbf{x}_k \boldsymbol{\theta}_S^*]$$

The trajectory of the shape particle is unaltered; only its dimensions are changed now without trying to keep the proportions.

Choosing again a MH method, the ratio of the acceptance probability of the move is: $\rho(\Lambda, \Lambda^*) = \frac{p(\Lambda^*|y_{1:k})}{p(\Lambda|y_{1:k})}$. It can be straightforwardly developed:

$$\rho(\Lambda, \Lambda^*) = \frac{\left[\prod_{i=1}^k p(y_i|\mathbf{x}_i, \boldsymbol{\theta}_S^*)\right] \cdot p(\boldsymbol{\theta}_S^*)}{\left[\prod_{i=1}^k p(y_i|\mathbf{x}_i, \boldsymbol{\theta}_S)\right] \cdot p(\boldsymbol{\theta}_S)} \tag{18}$$

Such a move would require evaluating all the likelihood terms corresponding to all the past measurements, leading to computational and memory requirements increasing in time. And yet, the deformation move K_D can be easily evaluated at the time of the 1st measurement, where:

$$\rho(\Lambda,\Lambda^*)=\frac{p(y_1|\mathbf{x}_1,\theta_S^*)\cdot p(\theta_S^*)}{p(y_1|\mathbf{x}_1,\theta_S)\cdot p(\theta_S)} \tag{19}$$

Moreover it can be combined with the transition move K_T. Then the SAPF behaves as a simulated annealing importance sampling. For the next measurements, an artificial Gaussian diffusion of deformation is applied through the successive layers of the SAPF.

5. Results

In this section, we present shape video results, starting from the simulated annealing behaviour to 3D tracking, attribute estimation and joint tracking and identification.

5.1. Simulated annealing behaviour

The simulated annealing layers of the SAPF (500 particles) are illustrated in Fig. 8 (for the 1st measurement). Let $\mathscr{U}(A)$ denote the uniform distribution on the set A. The dimensions of the parallelepiped are unknown ($l,w,h\sim\mathscr{U}$[10 cm 30 cm]), the position and orientation of the target are also unknown: the range $p(r_0)=\mathscr{U}$[1 m 2 m], image plane coordinates $p(x_0,y_0)$ is deduced from the 1st measurement and the three Euler angles are completely undetermined ($\sim\mathscr{U}[0^\circ\ 360^\circ]$). The adaptive cooling schedule is determined by allowing the effective sample size to decrease by 30% at each step.

The upper left image represents the prior knowledge: all the parallelepiped centres are drawn (red points) and the edges of five of them, randomly chosen, are added. In the next images, only the edges associated with the MMSE estimate is represented. After six layers, the true mode has been approximately found. Afterwards, the particles migrate to the shape centre. Finally, the posterior distribution is concentrated on the true centre and the MMSE is perfectly superposed on the parallelepiped image; the SAPF satisfactorily estimates the shape position, orientation and dimensions.

5.2. Shape 6D tracking

Fig. 9 represents the tracking of a hemisphere during 75 measurements at the frame rate of 15 Hz. The radius of the hemisphere is known (20 cm) and the prior, represented in the upper left im-

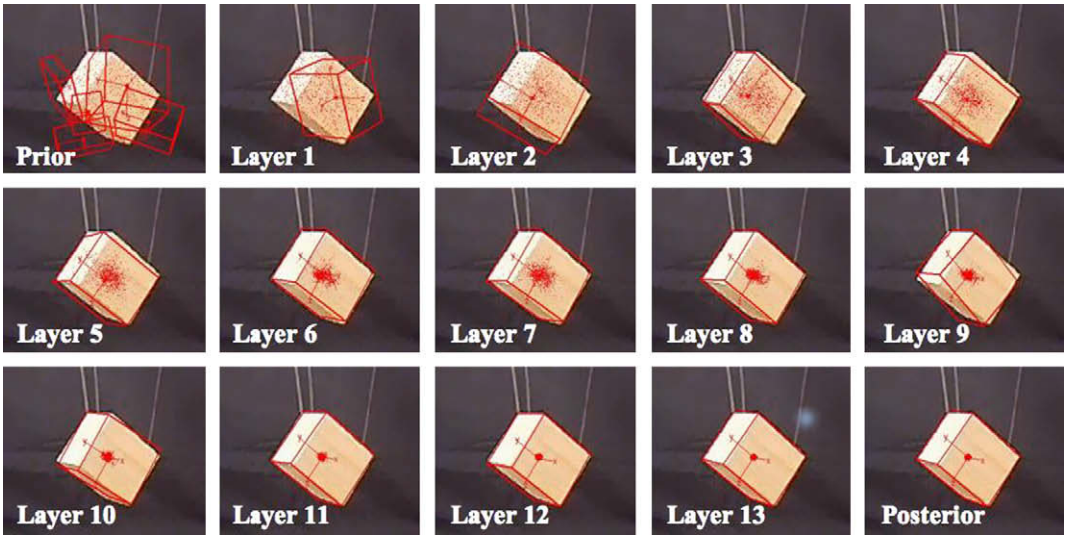

Fig. 8. SAPF (1st measurement – 500 particles).

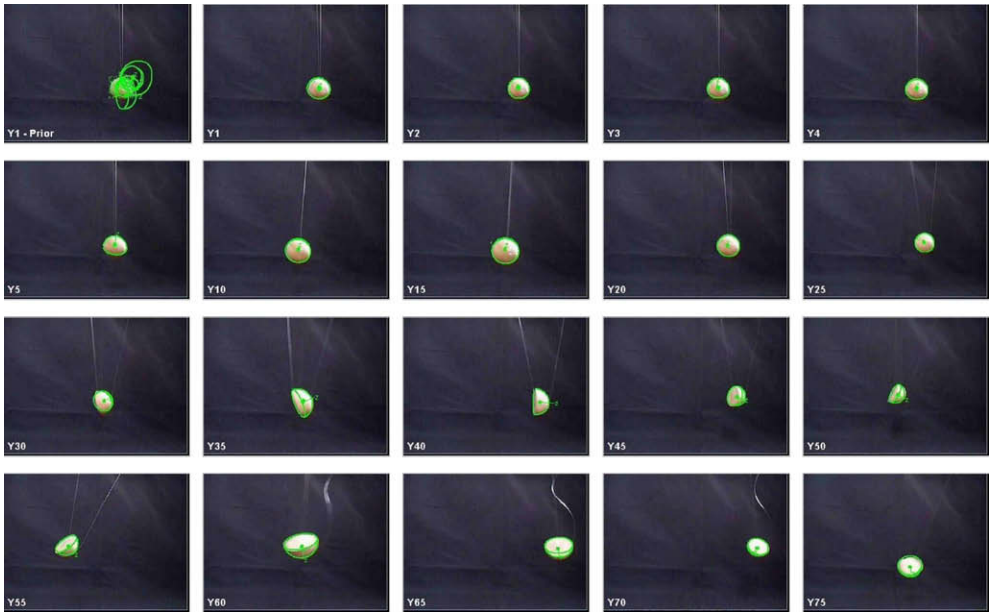

Fig. 9. Image sequence of the hemisphere tracking.

age, is similar to the previous illustration, with each velocity component $\sim \mathcal{U}[-1\ \text{m/s}\ 1\ \text{m/s}]$. For a while, the target moves slowly and only shows its spherical face; after the 25th measurement, it moves faster while rotating. Overall the SAPF is able to track the target satisfactorily. Sometimes, slight imperfections occur when the extracted features of the image are mediocre, i.e., mainly when the edges are not well located (measurements Y_{30}, Y_{50} and Y_{65}). Around Y_{35}, the tracker has not found the true orientation. Subsequently, it manages to find the true mode again. Fig. 10 shows the trajectory of the target centre in the normal plane OXZ (from above). The uniform prior is represented in red, the successive MMSE estimates of the position are drawn with pink circles and, for a few measurements, the particle clouds are represented with blue points.

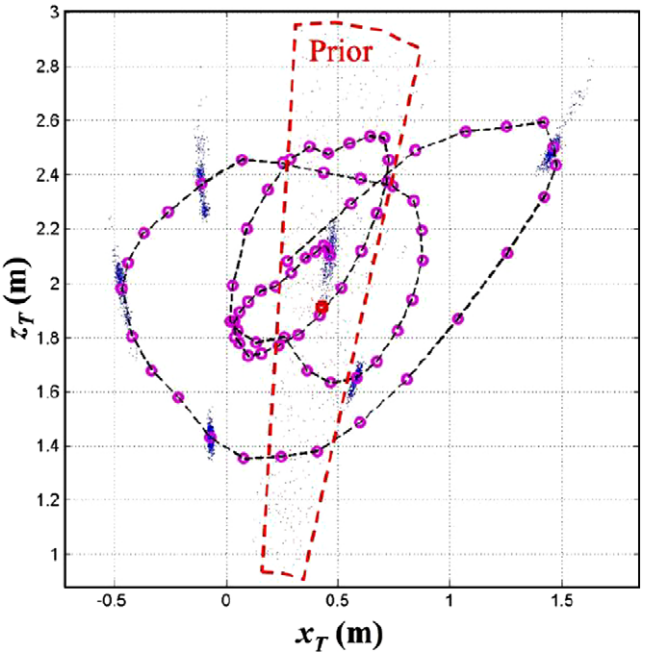

Fig. 10. Hemisphere trajectory.

Concerning the evolution of the number of layers N_{Layer}, controlled by the adaptive cooling schedule, the 1st inference requires 10 layers. Afterwards, N_{Layer} oscillates between 5 and 6 for a few tens of measurements. It increases substantially after time 30 while the target motion is more pronounced and the prior distribution $p(x_n|x_{n-1})$ further from the likelihood.

5.3. Estimation of shape dimensions

5.3.1. 1D attribute

The first example deals with the estimation of a mono-dimensional parameter, the radius of a moving sphere. The prior and the associated cloud of 100 particles are represented in the upper left part of Fig. 11, with the range $r_0 \sim \mathcal{U}[5\ \text{m}\ 10\ \text{m}]$ and the radius $r \sim \mathcal{U}[20\ \text{cm}\ 100\ \text{cm}]$. After, the set of samples approximating the posterior is represented at time 1, 5 and 10. The corresponding images of the moving sphere and the MMSE estimators are also represented. As soon as the full MCMC kernel is used on the first measurement, the radius is estimated quite accurately for a given range and there is a strong correlation between the range and the radius (meaning the target could be further and bigger). Black circles represent a possible trajectory of a 60-cm radius sphere

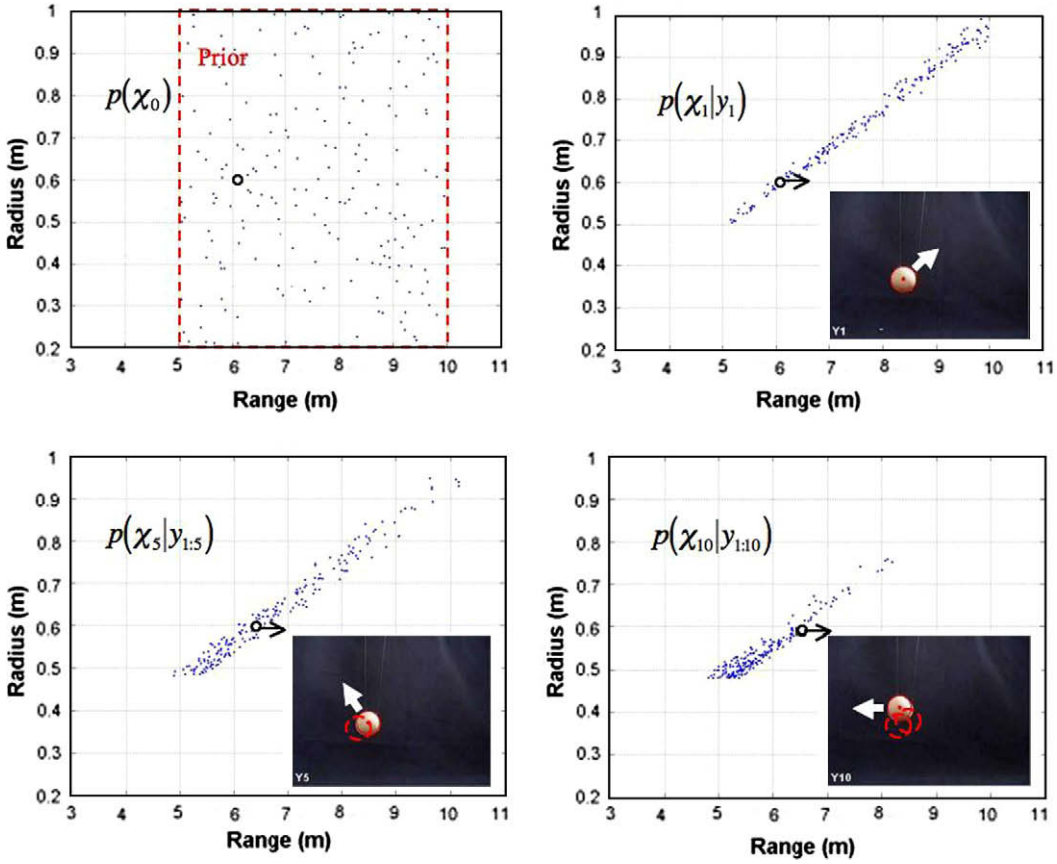

Fig. 11. Sphere tracking with unknown radius.

moving away from the camera. The dynamic model favours closer trajectories and the distribution becomes increasingly localised little by little. The relative spread in range at Y_5 and Y_{10} is due to a degradation of the image quality. Note that this estimation does not require the deformation move K_D; the re-scaling move K_λ is sufficient.

5.3.2. 3D hyper-parameter

The second example deals with the estimation of a 3D parameter, i.e., the length (l), the width (w) and the height (h) of a parallelepiped. The prior distributions on the three dimensions l, w and h are similar: $\mathcal{U}$[10 cm 30 cm]. The proportions are represented in the upper left part of Fig. 12, in the plane $(w^*/l^*, h^*/l^*)$ where l^*, w^* and h^* are the sorted dimensions in a decreasing order. Black circles represent the true proportion while green squares represent the successive MMSE estimates. After the first full MCMC move, the set of samples tightens around the true point. The next hybrid inference, based on MCMC and an artificial diffusion, gradually improves the estimate. At time 20, the estimate is within a few percents of the true value. Subsequently, the estimation deteriorates due to the likelihood being biased by an imperfect suppression of the threads and a resulting incorrect evaluation of the mismatched edges.

5.4. Joint tracking and identification

5.4.1. Scenario I – precise classes (sphere/hemisphere/cube)

In this first scenario, the measurements are the 60 first measurements of the previous hemisphere 6D tracking example. There are now three equiprobable classes ($p_i = 1/3$), the shape dimensions in each class being completely known: *sphere*($r = 20$ cm)/*hemisphere*($r = 20$ cm)/*cube*($l = 30$ cm).

Fig. 13 displays the evolution of the one-step ahead predictive densities, i.e., $p(y_n|y_{1:n-1}, i)$, and of the class posterior probabilities. The predictive densities of the classes sphere and hemisphere are very close to each other until time 28. Afterwards, while the target rotates, the predictive density of the class sphere collapses and it becomes highly probable that the target is in fact a hemisphere. On the other hand, the cube class seems immediately unlikely; it only could be a possible alternative around time 50 (and only at those time instants) when the target is quite far from the camera and the edges are badly extracted. Note that for all the classes the predictive densities increase significantly between the 1st and the 2nd measurement since the 1st image reduces dramatically the uncertainty and $p(y_2|y_1)$ is consequently far narrower than $p(y_1)$.

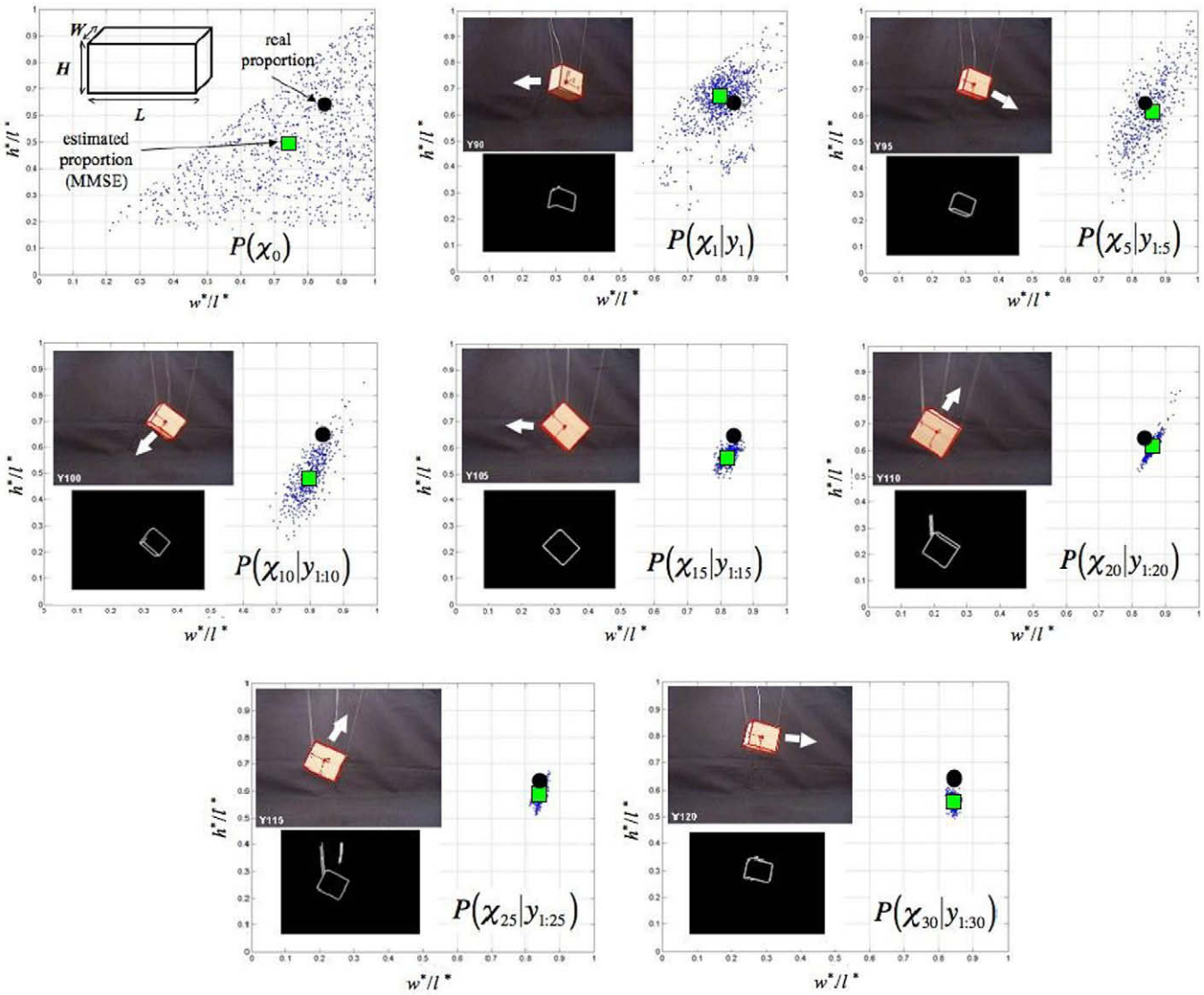

Fig. 12. Parallelepiped tracking with uncertain proportions in the three dimensions.

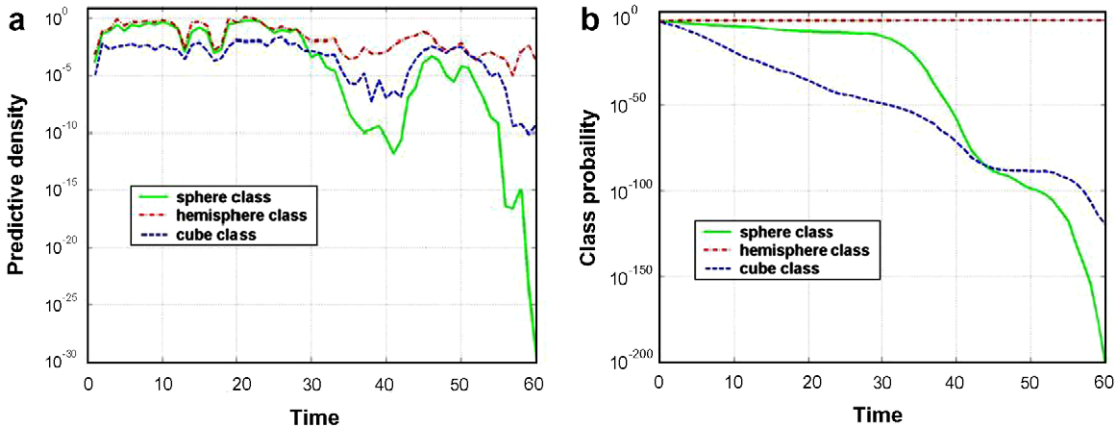

Fig. 13. Scenario I – class predictive density (a) and class probability (b).

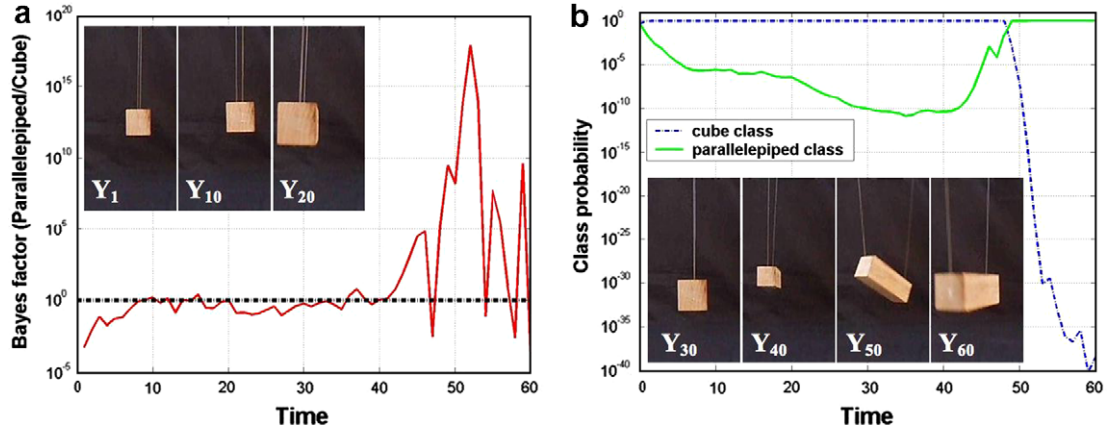

Fig. 14. Scenario II – Bayes factor (a) and class probability (b).

5.4.2. Scenario II – imprecise classes (cube/parallelepiped)

In this second scenario, the real target is the long parallelepiped. For a while, it only shows its smallest square section, then rotates and turns its longest face towards the camera from around time 40. There are now two equiprobable ($p_i = 1/2$) classes: *cube* ($r \sim \mathcal{U}[10\text{ cm}\ \ 30\text{ cm}]$)/*parallelepiped* ($l = 39.5\,\text{cm} - w,\ h \sim \mathcal{U}[10\text{ cm}\ 30\text{ cm}]$).

Fig. 14 represents the evolution of the Bayes factor and the class posterior probabilities. Until time 40, the class cube has the highest posterior probability, essentially due to the fact that rotating the long parallelepiped would typically have shown its other faces more quickly. After time 40, it becomes more evident that it is actually a long parallelepiped. Notice the few jolts of the Bayes factor that are the result of poor edge extractions associated with some specific frames.

5.5. Computational requirements

The shape video software, developed in Matlab and C, was run on a PC Pentium II (350 MHz). Using 200 particles and 5–6 annealing layers on average, a typical sequence of 15 s (225 frames – 15 Hz – image resolution 320×240 pixels) takes approximately 2 h. However, the code is not optimised; better tunings of the SAPF and improvements of the numerical likelihood evaluation routines would allow to speed up significantly the code.

6. Conclusion

In this paper, we have proposed a Bayesian approach to address the joint target tracking and identification problem. Although a Bayesian approach to this problem is conceptually straightforward, it is difficult to solve the associated computational problems when the candidates models include static parameters. We have reviewed the limitations of standard approaches and have proposed a sophisticated SMC method based on MCMC moves to perform inference. This methodology has been demonstrated successfully on the tracking and identification of geometric shapes in video sequences.

Acknowledgements

This research was completed by the French MOD – DGA/DSFP, Contract No. 02.60.00.060 and was completed at QinetiQ Malvern.

References

[1] D.L. Hall, J. LLinas (Eds.), Handbook of Multisensor Data Fusion, CRC Press, Berlin, 2001.
[2] S. Blackman, R. Popoli, Design and Analysis of Modern Tracking Systems, Artech House, 1999.
[3] A. Doucet, J.F.G. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001.
[4] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Stat. Comput. 10 (3) (2000) 197–208.
[5] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.
[6] N. Gordon, S. Maskell, T. Kirubarajan, Efficient particle filters for joint tracking and classification, in: SPIE Signal and Data Processing of Small Targets 2002, Orlando, April 2002.
[7] S. Maskell, Joint tracking of manoeuvring targets and classification of their manoeuvrability, EURASIP J. Appl. Signal Process. 2004 (15) (2004) 2339–2350.
[8] Y. Boers, H. Driessen, Integrated tracking and classification: an application of hybrid state estimation, in: SPIE Signal and Data Processing of Small Targets 2001, Orlando, April 2001.

[9] S. Challa, G. Pulford, Joint target tracking and classification using radar and ESM sensors, IEEE Trans. Aerospace Electron. Syst. 37 (3) (2001) 1039–1055.
[10] S. Herman, P. Moulin, A particle filtering approach to passive radar tracking and automatic target recognition, in: IEEE Aerospace Conference 2002, Big Sky, March 2002.
[11] C. Andrieu, A. Doucet, W.J. Fitzgerald, J.-M. Pérez, Bayesian Computational Approaches to Model Selection Nonlinear and Non-Gaussian Signal Processing, Newtown Institute Series, Cambridge University Press, Cambridge, 2000.
[12] C. Andrieu, N. de Freitas, A. Doucet, Sequential MCMC for Bayesian model selection, in: Proc. Work. IEEE HOS, 1999.
[13] C. Berzuini, W. Gilks, Resample-move filtering with cross-model jumps, in: A. Doucet, J.F.G. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001, pp. 117–138.
[14] P. Fearnhead, MCMC, sufficient statistics and particle filters report, J. Comput. Graph. Stat. 11 (4) (2002) 848–862.
[15] D.S. Lee, N.K.K. Chia, A particle algorithm for sequential bayesian parameter estimation and model selection, IEEE Trans. Signal Process. 50 (2) (2002) 326–336.
[16] J. Liu, M. West, Combined parameter and state estimation, in: A. Doucet, J.F.G. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001, pp. 197–217.
[17] G. Storvik, Particle filters for state-space models with the presence of unknown static parameters, IEEE Trans. Signal Process. 50 (2) (2002) 281–289.
[18] S. Godsill, T. Clapp, Improvement strategies for Monte Carlo particle filters, in: A. Doucet, J.F.G. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001, pp. 139–158.
[19] T. Clapp, Statistical Methods in the Processing of Communications Data, Ph.D. thesis, Cambridge University Engineering Department, 2000.
[20] R.M. Neal, Annealed importance sampling, Stat. Comput. 11 (2) (2001) 125–139.
[21] J. Deutscher, A. Blake, I. Reid, Articulated body motion capture by annealed particle filtering, in: IEEE Conference on Computer Vision and Pattern Recognition 2000, Hilton Head Island, June 2000.
[22] C. Berzuini, N.G. Best, W.R. Gilks, C. Larizza, Dynamic conditional independence models and Markov chain Monte Carlo methods, J. Am. Stat. Assoc. 92 (440) (1997) 1043–1141.
[23] C. Andrieu, N. de Freitas, A. Doucet, M.I. Jordan, An introduction to MCMC for machine learning, Mach. Learn. 50 (2003) 5–43.
[24] Y. Bar-Shalom, X.R. Li, Estimation and Tracking – Principles Techniques and Software, Artech House, 1993.
[25] P. Del Moral, A. Doucet, A. Jasra, Sequential Monte Carlo samplers, J. R. Stat. Soc. B 68 (3) (2006) 411–436.
[26] D.B. Reid, An algorithm for tracking multiple targets, IEEE Trans. Autom. Control 24 (6) (1979) 843–854.
[27] D.L. Alspach, A Gaussian sum approach to the multi-target-identification-tracking problem, Automatica 11 (1975) 285–296.
[28] D. Crisan, A. Doucet, A survey of convergence results on particle filtering methods for practitioners, IEEE Trans. Aerospace Electron. Syst. 50 (3) (2002) 736–746.