Parallel Subspace Sampling for Particle Filtering in Dynamic Bayesian Networks

Eva Besada-Portas[1], Sergey M. Plis[1], Jesus M. de la Cruz[2], and Terran Lane[1]

[1] University of New Mexico, Albuquerque NM 87131, USA
[2] Universidad Complutense de Madrid, 28040 Madrid, Spain

Abstract. Monitoring the variables of real world dynamic systems is a difficult task due to their inherent complexity and uncertainty. Particle Filters (PF) perform that task, yielding probability distribution over the unobserved variables. However, they suffer from the curse of dimensionality problem: the number of particles grows exponentially with the dimensionality of the hidden state space. The problem is aggravated when the initial distribution of the variables is not well known, as happens in global localization problems. We present a new parallel PF for systems whose variable dependencies can be factored into a Dynamic Bayesian Network. The new algorithms significantly reduce the number of particles, while independently exploring different subspaces of hidden variables to build particles consistent with past history and measurements. We demonstrate this new PF approach on some complex dynamical system estimation problems, showing that our method successfully localizes and tracks hidden states in cases where traditional PFs fail.

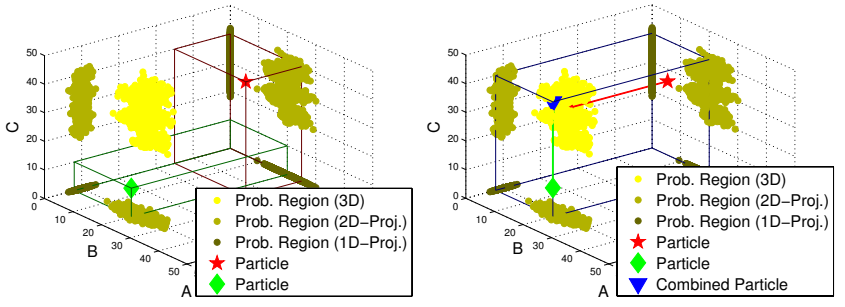
1 Introduction

Estimating the hidden state of a multivariate dynamical system remains a large challenge. While the Dynamic Bayesian Network (DBN) formalism provides us an excellent way to *represent* such systems, performing *inference* in these models remains difficult, and approximations are usually necessary. Among the most popular approximate inference methods for DBNs are *particle filters* (PFs): point-mass approximations to the hidden state distribution, which are updated via Monte Carlo sampling steps [1].

In spite of their popularity, PFs can be sensitive and tricky to apply to new problems. One of the core challenges arises from the familiar curse of dimensionality: in even modest dimensionality spaces, the chances are high that many or most particles will fall into near-zero probability regions of the state space, leading to serious particle depletion and quickly driving the PF off track. This problem is evident, for example, in multi-object tracking tasks [2]. The difficulty is exacerbated by poor initial particle distributions that are unlikely to choose any high-probability particles. The dimensionality/depletion difficulty can be reduced by careful choice of initial particle distribution and sampling and reweighting distributions, but doing so requires extensive domain knowledge engineering.

In this paper, we propose a principled sampling framework to overcome some of the dimensionality drawbacks for PFs applied to DBNs with factored state spaces and multiple conditionally independent observations (e.g. multi-object tracking). Our approach

W. Buntine et al. (Eds.): ECML PKDD 2009, Part I, LNAI 5781, pp. 131–146, 2009.



(a) Low probability (b) High probability

Fig. 1. Combining the information of two low probability particles (Figure 1a) to create a higher probability one (Figure 1b)

requires less prior knowledge about domain dynamics and fewer ad hoc corrections than other methods [3,4].

The key insight is displayed in Figure 1a. Here we see a state space of three variables, $\{A,B,C\}$, with the support of the target distribution displayed as a bright yellow cloud centered at approximately $[A=40,B=20,C=30]$. Two particles (red star and green diamond) fall far from the target distribution and are assigned zero probability. However, the red star particle has significant probability in the projection into the $\{C\}$ subspace (dark olive cloud), while the green diamond particle has non-negligible probability in the A,B subspace (dark olive cloud). Our method amounts to carefully combining the useful dimensions of these particles (A and B from the green diamond particle and C from the red star) into a hybrid particle (Figure 1b, blue triangle) that falls within our target distribution. The trick is doing so in a way that correctly accounts for both the observation likelihoods associated with different subspaces and the particles' previous state histories.

Of course, the basic insight is not, itself, new [5,6,7,8]. What we add to previous work is a principled design of the PF importance sampling (IS) and weighted resampling (WR) steps that provides a firm probabilistic foundation for combining such particles. Doing so has three benefits. First, it allows a form of "parallel" filtering that works with separate, lower-dimensional subspaces of variables simultaneously (Section 3). Second, it helps overcome poor particle initialization, improving state localization and convergence when the initial particle sample is far from the target distribution (Section 5). And finally, it allows us to get away with simultaneously fewer particles and less prior knowledge/less sophisticated sampling distributions than existing methods (Section 4).

2 Particle Filter Fundamentals

Our core contribution is to modify the standard PF for DBNs [9] by substituting its proposal distribution for another that lets it sample the subspaces defined for different subsets of hidden variables and introducing the weight update equation related with the new proposal. In this section we introduce an uniform notation and formalism in which

to state the standard PF and our modifications to it. In the process, we also introduce a sub-contribution: the "instantaneous" PF, a variation of the standard PF that emerges to estimate only the current state of the hidden variables instead of the state of all the hidden variables at any time.

2.1 Definitions and Notation

In this paper, a capital letter U represents a random variable, a boldface capital letter $\boldsymbol{U}$ – a set of random variables, a lowercase letter u – the specific value of the corresponding random variable U, and a lowercase bold letter $\boldsymbol{u}$ – an assignment of the values to the variables of its set $\boldsymbol{U}$. We also define $\mathcal{P}(\boldsymbol{U})$ as a partition of $\boldsymbol{U}$.

A BN is an annotated directed graph that encodes the probability distribution of a set of random variables $\boldsymbol{V}$. DBNs model the distribution of a sequence of sets of variables. A set of variables belonging to k^{th} time slice is represented as $\boldsymbol{V}_k$ and the total set of variables up to the current time slice t is $\boldsymbol{V}_{0:t}$. To distinguish hidden and observed variables, we use X, $\boldsymbol{X}_t$ and $\boldsymbol{X}_{0:t}$ to denote the former, and Y, $\boldsymbol{Y}_t$ and $\boldsymbol{Y}_{0:t}$ for the latter. The graph is completely defined by the sets of parents of all its variables: $\mathrm{Pa}_k(V)$ represents the subset of the parents of variable V that belong to time slice k and $\mathrm{pa}_k(V)$ their assignment to particular values. Similarly, we also define the set of children of a variable and their assignments by $\mathrm{Ch}_k(V)$ and $\mathrm{ch}_k(V)$.

Probability distributions will be represented as $\mathrm{p}(\cdot)$, $\mathrm{q}(\cdot)$ and $\mathrm{r}(\cdot)$: the first related with the probabilities of the variables of the system and the others with the proposal distributions used to sample the particles from. The operation $a \sim \mathrm{q}(\cdot)$ represents sampling a according to $\mathrm{q}(\cdot)$.

A PF approximates the probability $\mathrm{p}(\boldsymbol{X}|\boldsymbol{y})$ of a set of hidden variables $\boldsymbol{X}$ given the values of the measurements $\boldsymbol{Y}$ by the point mass distribution $\sum_{i=1}^{N} w^{(i)} \delta(\boldsymbol{X} - \boldsymbol{x}^{(i)})$, where $\delta(\cdot)$ is Dirac delta function, $\boldsymbol{x}^{(i)}$ are the values of the variables in $\boldsymbol{X}$ in the i-th particle, $w^{(i)}$ their weights, and N the number of particles. Additionally, $x^{(i)}$, $\boldsymbol{x}_{0:t}^{(i)}$ and $\mathrm{pa}_t^{(i)}(V)$ represent the assignments of variable X, the variables in $\boldsymbol{X}_{0:t}$ and the variables in $\mathrm{Pa}_t(V)$ to the values they have in the i-th particle, and $w_t^{(i)}$ stands for the weight of the particle when we have unrolled the DBN up to the time stamp t.

2.2 Importance Sampling (IS) and Weighted Resampling (WR)

To develop a new PF we can modify the two basic operations that are normally combined to obtain the values of the particles $\boldsymbol{x}^{(i)}$ and the weights $w^{(i)}$ [10]:

- Importance sampling is used to (1) create new particles by means of a proposal distribution $\mathrm{q}(\boldsymbol{X}|\boldsymbol{y})$ that generates their values ($\boldsymbol{x}^{(i)} \sim \mathrm{q}(\boldsymbol{X}|\boldsymbol{y})$) and (2) calculate their weights as $w^{(i)} = \mathrm{p}(\boldsymbol{x}^{(i)}|\boldsymbol{y}) / \mathrm{q}(\boldsymbol{x}^{(i)}|\boldsymbol{y})$.
- Weighted resampling is used to redistribute an existing set of weighted particles $(\boldsymbol{x}^{(i)}, w^{(i)})$ according to the resampling function $\mathrm{r}(\boldsymbol{X})$, without changing the approximated distribution. The new set of weighted particles $(\boldsymbol{x}'^{(i)}, w'^{(i)})$ is obtained as $\boldsymbol{x}'^{(i)} = \boldsymbol{x}^{(j)}$ and $w'^{(i)} = w^{(j)} / \mathrm{r}(\boldsymbol{x}^{(j)})$ with $j \sim r(\boldsymbol{x}^{(j)})$.

In short, IS is used to create the particles while WR is carried out to increment the number of the particles in the regions of high interest and reduce them in the others.

2.3 PF for DBN

Note that we have not specified which random variables are included in the sets $\boldsymbol{X}$ and $\boldsymbol{Y}$. Although it is often non stated explicitly in the literature, different PFs emerge from different choices of hidden variables $\boldsymbol{X}$ and observed values $\boldsymbol{y}$. When $\boldsymbol{X}=\boldsymbol{X}_{0:t}$ the PF is responsible of estimating the probability of the trajectory $\boldsymbol{X}_{0:t}$ while when $\boldsymbol{X}=\boldsymbol{X}_t$ is in charge of estimating the probability of the state $\boldsymbol{X}_t$ at the current time slice t. To distinguish the two problems, we name the PF filters that solve the first problem "trajectory" PF and the second "instantaneous". The set of observed variables $\boldsymbol{Y}$ is in both cases $\boldsymbol{Y}=\boldsymbol{Y}_{0:t}$.

The IS operation of both filters can be carried out sequentially exploiting the independence assumption imposed by the structure of the DBN. In the paper, we only consider DBNs whose variables have parents only belonging to the current or previous time slice ($\forall V\in\boldsymbol{V}_t \wedge \forall k\notin\{t,t-1\}\ \mathrm{Pa}_k(V)=\emptyset$). With this restriction, the structure of the DBN factors the joint probability of the set of hidden and observation variables up to time slice t as in Eq. (1). The dependence of each variable on its parents imposes an ancestral ordering in the evaluation of the probabilities that needs to be considered by the PFs.

$$\begin{aligned}\mathrm{p}(\boldsymbol{X}_{0:t},\boldsymbol{Y}_{0:t}) &= \mathrm{p}(\boldsymbol{X}_t,\boldsymbol{Y}_t|\boldsymbol{X}_{0:t-1},\boldsymbol{Y}_{0:t-1})\,\mathrm{p}(\boldsymbol{X}_{0:t-1},\boldsymbol{Y}_{0:t-1})\\ \mathrm{p}(\boldsymbol{X}_t,\boldsymbol{Y}_t|\boldsymbol{X}_{0:t-1},\boldsymbol{Y}_{0:t-1}) &= \prod_{X\in\boldsymbol{X}_t}\mathrm{p}(X|\mathrm{Pa}_{t-1}(X),\mathrm{Pa}_t(X))\prod_{Y\in\boldsymbol{Y}_t}\mathrm{p}(Y|\mathrm{Pa}_{t-1}(Y),\mathrm{Pa}_t(Y))\end{aligned} \quad (1)$$

Although the two types of PFs work with different $\boldsymbol{X}$, both are used to obtain the values of the current state variables $\boldsymbol{X}_t$, because the trajectory $\boldsymbol{X}_{0:t}$ includes them. Additionally, for some types of proposals $\mathrm{q}(\boldsymbol{X}|\boldsymbol{y})$, they are equivalent. However, the search space of the first is significantly bigger, and it usually needs more particles, as [11] shows for the case of HMMs.

"Trajectory" PF. Its weights $w_t^{(i)}$ can be calculated with expression (2), which exploits the factorization of the DBN, assumes that $\mathrm{q}(\boldsymbol{x}_{0:t-1}^{(i)}|\boldsymbol{y}_{0:t})=\mathrm{q}(\boldsymbol{x}_{0:t-1}^{(i)}|\boldsymbol{y}_{0:t-1})$ and considers that instead of sampling from $\mathrm{q}(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{0:t-1}^{(i)},\boldsymbol{y}_{1:t})$ we can do it from $\mathrm{q}(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t})$.

$$\begin{aligned}w_t^{(i)} &= \frac{\mathrm{p}\left(\boldsymbol{x}_{0:t}^{(i)}|\boldsymbol{y}_{0:t}\right)}{\mathrm{q}\left(\boldsymbol{x}_{0:t}^{(i)}|\boldsymbol{y}_{0:t}\right)} \propto \frac{\mathrm{p}\left(\boldsymbol{x}_{0:t}^{(i)},\boldsymbol{y}_{0:t}\right)}{\mathrm{q}\left(\boldsymbol{x}_{0:t}^{(i)}|\boldsymbol{y}_{0:t}\right)} = \frac{\mathrm{p}\left(\boldsymbol{x}_t^{(i)},\boldsymbol{y}_t|\boldsymbol{x}_{0:t-1}^{(i)},\boldsymbol{y}_{0:t-1}\right)}{\mathrm{q}\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{0:t-1}^{(i)},\boldsymbol{y}_{0:t}\right)}\frac{\mathrm{p}\left(\boldsymbol{x}_{0:t-1}^{(i)},\boldsymbol{y}_{0:t-1}\right)}{\mathrm{q}\left(\boldsymbol{x}_{0:t-1}^{(i)}|\boldsymbol{y}_{0:t-1}\right)}\\ &\propto \frac{\prod_{Y\in\boldsymbol{Y}_t}\mathrm{p}\left(y|\mathrm{pa}_{t-1}^{(i)}(Y),\mathrm{pa}_t^{(i)}(Y)\right)\prod_{X\in\boldsymbol{X}_t}\mathrm{p}\left(x^{(i)}|\mathrm{pa}_{t-1}^{(i)}(X),\mathrm{pa}_t^{(i)}(X)\right)}{\mathrm{q}\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t}\right)}w_{t-1}^{(i)}\end{aligned} \quad (2)$$

Different proposals and combinations of IS and WR create different "trajectory" PF. The IS step of KLPF [9] uses $\mathrm{q}(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t})=\prod_{X\in\boldsymbol{X}_t}\mathrm{p}(x^{(i)}|\mathrm{pa}_{t-1}^{(i)}(X),\mathrm{pa}_t^{(i)}(X))$, and so $w_t^{(i)}=\prod_{Y\in\boldsymbol{Y}_t}\mathrm{p}(y|\mathrm{pa}_{t-1}^{(i)}(Y),\mathrm{pa}_t^{(i)}(Y))w_{t-1}^{(i)}$. According to the ancestral ordering of the variables, for each $X\in\boldsymbol{X}_t$ its value $x^{(i)}\sim\mathrm{p}(X|\mathrm{pa}_{t-1}^{(i)}(X),\mathrm{pa}_t^{(i)}(X))$ and for each $Y\in\boldsymbol{Y}_t$ multiplies $w_t^{(i)}$ by its corresponding likelihood $\mathrm{p}(y|\mathrm{pa}_{t-1}^{(i)}(Y),\mathrm{pa}_t^{(i)}(Y))$. The WR step of KLPF

is carried out once all $V \in \boldsymbol{V}_t$ are processed, using $\mathrm{r}(\boldsymbol{x}^{(i)}) = w_t^{(i)}$, what makes the resampled particles have all the same weight. The PF in [12] uses the same factorization and sequential weight updates as KLPF, but it alternates, according to the ancestral ordering, IS for each $X \in \boldsymbol{X}_t$ and WR with $\mathrm{r}(\boldsymbol{x}^{(i)}) = \mathrm{p}(y|\mathrm{pa}_{t-1}^{(i)}(Y), \mathrm{pa}_t^{(i)}(Y))$ after the likelihood of each measurement $Y \in \boldsymbol{Y}_t$ is used to multiply the $w_t^{(i)}$. That is, KLPF updates completely $w_t^{(i)}$ taking into account all the variables of time slice t and does WR after it, while the PF in [12] updates the $w_t^{(i)}$ for each variable and performs WR after each measurement appears in the ancestral ordering. The other "trajectory" PFs listed in Section 1 combine KLPF with other techniques.

"Instantaneous" PF. To the best of authors' knowledge, there are no "instantaneous" PFs designed for general DBNs. However, its $w_t^{(i)}$ can also be calculated by extending the ideas that [11] utilizes for HMMs to the general DBN case. Equation (3), our first contribution in this paper, calculates the weights by exploiting the factorization of the DBN and marginalizing the hidden variables of the previous time slice. To simplify the expression, it is possible to take out of the summation those variables that don't have hidden parents belonging to the previous time slice ($V \in \boldsymbol{V}_t$ s.t. $\mathrm{Pa}_{t-1}(V) \cap \boldsymbol{X}_{t-1} = \emptyset$).

$$
\begin{aligned}
w_t^{(i)} &= \frac{\mathrm{p}\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{y}_{0:t}\right)}{\mathrm{q}\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{y}_{0:t}\right)} \propto \frac{\mathrm{p}\left(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t|\boldsymbol{y}_{0:t-1}\right)}{\mathrm{q}\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{y}_{0:t}\right)} \propto \frac{\int \mathrm{p}\left(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t, \boldsymbol{x}_{t-1}|\boldsymbol{y}_{0:t-1}\right) d\boldsymbol{x}_{t-1}}{\mathrm{q}\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{y}_{0:t}\right)} \\
&\propto \frac{\int \mathrm{p}\left(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{0:t}\right) \mathrm{p}(\boldsymbol{x}_{t_1}|\boldsymbol{y}_{0:t-1})\, d\boldsymbol{x}_{t-1}}{\mathrm{q}\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{y}_{0:t}\right)} \\
&\propto \frac{\sum_{j=1}^{N}\left(w_{t-1}^{(j)} \prod_{Y \in \boldsymbol{Y}_t} p\left(y|\mathrm{pa}_{t-1}^{(j)}(Y), \mathrm{pa}_t^{(i)}(Y)\right) \prod_{X \in \boldsymbol{X}_t} p\left(x^{(i)}|\mathrm{pa}_{t-1}^{(j)}(X), \mathrm{pa}_t^{(i)}(X)\right)\right)}{q\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{y}_{0:t}\right)}
\end{aligned}
\tag{3}
$$

Although a factorization of the proposal distribution is also possible, the sequential way proposed for updating $w_t^{(i)}$ in the "trajectory" PF proposed [12] is no longer valid for the hidden variables that have to stay inside the summation.

3 A Parallel Sampling Framework for DBN

In a case with N hidden variables where there is an observation for each of these variables, the likelihood of a state is a product of the likelihoods of each hidden variable in the state. A likely particle should be likely according to each and all of its variables, otherwise its weight is going to be negligible. An easy situation arises when all hidden variables are independent and their joint probability can be factored into a product of its marginals. In this case, we can simply solve N PF in parallel and obtain a solution. The problem arises when there are inter-dependencies in the hidden state, i.e. we have a DBN of N hidden interacting random variables and their N conditionally independent observations. If our prior is slightly wrong in even one of the hidden variables it is difficult to obtain a particle with a non-negligible weight. Thus a wrong start usually leads to the hidden state representation completely going off track.

Existing approaches to addressing the problem can be roughly split into three categories: 1) tracking only a subset of hidden variables while handling its complement through additional probabilistic techniques [3,4], 2) sequentially alternating IS and WR sub-steps for the variables belonging to each time slice of the DBN [10,12], and (3) parallel creation of particles in subsets of hidden variables with their subsequent recombination by varying means [5,6,7,8].

The PFs presented in Section 2.3, sample the values of the current time hidden variables $\boldsymbol{X}_t$ from the values that their parents have inside the same particle. So the probability of each particle is highly dependent on the values that all the variables have inside that particle and therefore on the initial sampling distribution. In problems with many hidden variables, if no particle is "good" initially or at some point, there is little chance for the presented PFs to recover.

However, the values of some variables of some particles that have zero probability can be inside of the non-zero probability subspace associated with those variables. Additionally, different particles can have different subsets of variables inside different non-zero probability subspaces. This scenario is illustrated by Fig. 1a, where the light yellow region represents the non-zero probability zone of a set of three variables $\{A,B,C\}$, the darker olive surfaces and lines its projections into the different possible subspaces, and the red star and green diamond the position of two particles with zero probability but whose $\{C\}$ and $\{A,B\}$ variables are respectively inside probable subspaces. The core idea of our PF framework consists on using the information of probable areas of different particles to build non-zero probable particles by parallel sampling different subspaces of hidden variables. Figure 1b shows with a blue triangle a probable particle obtained combining the probable regions of the red and green particles.

However, to create probabilistic consistent PFs we need to combine IS and WR steps, and select the $\mathrm{q}(\cdot)$ and $\mathrm{r}(\cdot)$ functions. As the main distinction between the weight update operation of the "trajectory" and "instantaneous" PFs appears in the numerator, we can also use the same proposal for both types of filters. So, the parallel PF framework presented in this paper is based on 1) the proposal presented in the following section, that independently samples in parallel different subsets of hidden variables, and 2) in the calculus of the $w_t^{(i)}$ of the particles considering this proposal and Eq. (2) or (3).

3.1 A Parallel Sampling Proposal

Independently sampling different subspaces of hidden variables belonging to each time slice $\boldsymbol{X}_t$ can be achieved factoring the proposal $\mathrm{q}(\boldsymbol{x}_t^{(i)}|\cdot,\boldsymbol{y}_{1:t})$ that appears in the denominator of Eq. (2) or (3) in as many proposals as subsets $\boldsymbol{X}$ exist in a given disjoint partition $\mathcal{P}(\boldsymbol{X}_t)$ of $\boldsymbol{X}_t$:

$$\mathrm{q}\left(\boldsymbol{x}_t^{(i)}|\cdot,\boldsymbol{y}_{1:t}\right) = \prod_{\boldsymbol{X}\in\mathcal{P}(\boldsymbol{X}_t)} \mathrm{q}\left(\boldsymbol{x}^{(i)}|\cdot,\boldsymbol{y}_{1:t}\right) \tag{4}$$

The sampling proposal of each subset has to select highly probable values of their corresponding variables given all the past information. This behavior can be approximated with a mixture model of the product of the transition priors of all the variables belonging to each subset. However, to increment the flexibility of our PF, we substitute the

transition prior by a proposal of each variable given its parents. This lets the user decide whether to use the transition prior ($\mathrm{q}(x|\mathrm{pa}_{t-1}^{(j)}(X),\mathrm{pa}_t^{(i)}(X)) = \mathrm{p}(x|\mathrm{pa}_{t-1}^{(j)}(X),\mathrm{pa}_t^{(i)}(X))$) or another distribution that will let the PF explore other regions of the space. The complete proposal is presented in Eq. (5), where $\alpha_j^{\boldsymbol{X}}$ stands for the weight of each component of the mixture related with the subspace defined by the variables in the subset $\boldsymbol{X}$.

$$\mathrm{q}\left(\boldsymbol{x}_t^{(i)}|\cdot,\boldsymbol{y}_{1:t}\right) = \prod_{\boldsymbol{X}\in\mathcal{P}(\boldsymbol{X}_t)}\left(\sum_{j=1}^{N}\left(\alpha_j^{\boldsymbol{X}}\prod_{X\in\boldsymbol{X}}\mathrm{q}\left(x|\mathrm{pa}_{t-1}^{(j)}(X),\mathrm{pa}_t^{(i)}(X)\right)\right)\right) \tag{5}$$

The selection of the $\alpha_j^{\boldsymbol{X}}$ is fundamental because it let us change the importance of each component of the mixture. To make it consistent with the past measurements we can make it proportional to the product of the likelihoods related with the hidden variables of the subspace defined by the subset $\boldsymbol{X}$. However, to increment the flexibility of the system, we substitute the likelihoods for other distributions that relate the observed variables with their parents as it is presented in equation (6).

$$\alpha_j^{\boldsymbol{X}} = \prod_{Y\in\mathrm{Ch}_t(X)\wedge X\in\boldsymbol{X}}\mathrm{q}\left(y|\mathrm{pa}_{t-1}^{(j)}(Y),\mathrm{pa}_t^{(i)}(Y)\right) \tag{6}$$

However, we can't sample from the mixture model if the weights depend on values of the different components of the mixture. To overcome this difficulty, we can follow the approach of Auxiliary PFs [13,11]: we will assign to each hidden variables $X\in\mathrm{Pa}_t(Y)$ in Eq. (6) the mean value that will be obtained from sampling from $\mathrm{q}(x|\mathrm{pa}_{t-1}^{(j)}(X),\mathrm{pa}_t^{(i)}(X))$ instead of a sampled one.

Equation (5) and (6) define completely our parallel proposal. However, it is important to highlight that the parallel sampling idea proposed by (4) can be extended to other types of factorization. For instance, we could extend the idea of substituting the transition priors and likelihoods by adding some extra components to the mixture that will let us sample from regions of space no related with any particle at the previous time slice or modifying $\alpha_j^{\boldsymbol{X}}$ to create more particles in certain regions of the space, according with some prior kwnowlegde of the problem.

3.2 Defining the Partition of the Hidden and Observation Variables

The structure of the DBN imposes some restriction on the disjoint partition of hidden variables. Additionally, Eq. (6) imposes some constraints too in the partition of the observed variables. Both are closely related.

For the partition of the hidden variables, the isochronal connected hidden variables and the isochronal hidden variables whose child is the same observed variable need to be in the same partition. That is, the basic hidden partition fulfills that $\forall\boldsymbol{X}\in\mathcal{P}(\boldsymbol{X}_t)B\in\boldsymbol{X}$ if $X\in\boldsymbol{X}\wedge(X\in\mathrm{Pa}_t(B)\vee(Y\in\boldsymbol{Y}_t\wedge B\in\mathrm{Pa}_t(Y)\wedge X\in\mathrm{Pa}_t(Y)))$. Any other partition is formed by the union of subsets of the basic one.

Regarding the partition of the observed variables, we need to divide them according to the selected partition of the hidden variables, grouping the measurements with its hidden parents. So, the basic observed partition fulfills $\forall\boldsymbol{Y}\in\mathcal{P}(\boldsymbol{Y}_t)\exists\boldsymbol{X}\in\mathcal{P}(\boldsymbol{X}_t)\mathrm{s.t.}\forall Y\in\boldsymbol{Y}\exists X\in\boldsymbol{X}\mathrm{s.t.}Y\in\mathrm{Ch}_t(X)$.

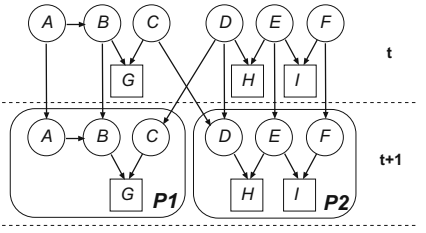


Fig. 2. Example of partition for a DBN

An example is presented in Fig. 2, where there is a DBN with 6 hidden variables $\{A,B,C,D,E,F\}$ and 3 measurements $\{G,H,I\}$ per time slice. This network has two basic hidden partitions (P1 and P2), represented with the big rounded squares. Variables A and B belong to P1 because $A \in \mathrm{Pa}_t(B)$. C also belongs to P1, because G is a common child of B and C. D, E and F belong to P2 because they are related through their common children H and I. The measurements are divided according to the hidden partitions: G belongs to the observed partition associated with P1 because it is a child of B and C, and H and I to the other because they are children of D, E and F.

3.3 The Complete Parallel Framework

The first step, before carrying out the filtering steps of our framework consists on defining the disjoint partitions, taking into account the restrictions presented in Sec. 3.2.

Once the subsets are defined, and the original particles created with an initialization proposal, we can sample the subspaces in parallel with the proposal defined by Eq. (5) and (6). In short, we get the mean values of the hidden variables and calculate the $\alpha_j^{\boldsymbol{X}}$ of each subset taking into account the ancestral ordering. We use $\alpha_j^{\boldsymbol{X}}$ to select a component of the mixture and sample the hidden values from that component.

To calculate the $w_t^{(i)}$ of the particles created with the parallel proposal, we use either (2) or (3). The numerator can be zero when the values of the hidden variables in the current time slice are non probabilistically "consistent" with the ones of the previous time slice and/or the measurements. The denominator can't, because we have generated it with the proposal. So, the final $w_t^{(i)} \geq 0$. When $w_t^{(i)} = 0$ for all the particles, as all of them are equally non probable, we accept them with the same probability. This automatically resets our PF to the values proposed by $\mathrm{q}(\cdot)$ when the PF is lost (all $w_t^{(i)} = 0$).

Additionally, we can also include a WR step after calculating $w_t^{(i)}$, or and create a Serial and Parallel sampling PF that alternate substeps of our Importance Parallel Sampling step with WR steps for the groups we sample in serial. However, the "instantaneous" PF can't include a WR step before we have sampled all the variables that can't be extracted from the numerator's summation in (3).

The computational cost of our PF framework depends on the expressions used for calculating the numerator ("trajectory" or "instantaneous" case) and denominator (the parallel sampling proposal) of (2) or (3). How to minimize the cost is out of the scope of this article, although we believe that the use of N-body learning can reduce it

significantly as implied by [11]. Additionally, we can use parallel computing architectures for calculating the probabilities values and sample groups of hidden variables.

4 Related Work

This section compares the behavior of our PFs with some of the existing ones, starting with the PFs that combine the values of different subsets of hidden variables to create new particles and following with other two techniques closely related with our PF. Table 1 highlights the most relevant parts of our comparison.

Among the first group of PFs, the Factored PF in [5] is developed for systems with discrete hidden variables. It approximates the distribution as a product of distributions of subsets of hidden variables. The probability of each subset is represented as a weighted set of particles, and the inference is carried out building particles of the whole set of hidden variables, performing a PF step with the whole particles, and marginalizing the whole PF distribution to obtain the point-mass distribution of each subset. Our PF, valid for DBNs with continuous and discrete variables, follows a different path: it keeps the weighted particle distribution of all hidden variables, samples from different subsets according to $\alpha_j^{\boldsymbol{X}}$, and builds the whole particle with the weights calculated by (2) or (3).

The Hybrid-PF for the specific problem of [6] combines the Factored PF for discrete variables with the Rao-Blackwellised technique for continuous ones, and includes a look-ahead step to sample the most probable particles according to the current measurements before building the whole particle. Our proposal is more general, doesn't distinguish discrete and continuous variables, and the look-ahead step is implicitly implemented with our selection of $\alpha_j^{\boldsymbol{X}}$.

The Genetic PF for HMM in [7] includes a Genetic Algorithm (GA) crossover and mutation step, placed after having sampled the values of the hidden variable with the transition prior and before calculating the weights with the version of (2) for HMM. This lets the PF explore the whole space, although the calculated $w_t^{(i)}$ don't consider the two new steps. So, its $w_t^{(i)}$ represents only the probability of a hidden variable at the current time given the current time measurement, although the probability of sampling areas of the space consistent with previous measurements is higher.

The Hierarchical PF in [8] consists in multiple PF steps connected in serial and in parallel. Although structurally the closest to our parallel PFs, it doesn't show how to update $w_t^{(i)}$ after the parallel connection, while our PFs provides the sampling distribution as well as the updating weights equations.

It is important to highlight the connection of our PFs with the "instantaneous" PF for HMM [11]. Its proposal, extended to DBNs, will be equal to ours when all the hidden and observed variables belong to the same partition $(\mathcal{P}(\boldsymbol{Y}_t) = \boldsymbol{Y}_t) \wedge (\mathcal{P}(\boldsymbol{X}_t) = \boldsymbol{X}_t)$.

Finally, the serial PF presented in [10,12], that sequentially alternates IS and WR sub-steps for the variables belonging to each time slice of the DBN, can be used in combination with our PFs to build general Serial/Parallel PFs because the two approaches complement each other: the serial approach is beneficial for DBNs which have many isochronal links of hidden variables and can only be partitioned into a few sets, while the parallel approach is beneficial for DBNs which are naturally partitioned in many

Table 1. Comparison of our PFs with others (D. Discrete, C. Continuous)

PF	Foundation	Variables	DBN type	Probabilistic Consistent weight update
Ours	Parallel sampling proposal	D. + C.	Highly parallel	Yes
[5]	Factored posterior + PF for particles created from the posterior	D.	Any	No
[6]	[5] + Rao-Blackwellised	D. + C.	Special	No
[7]	Normal PF + genetic mutation and crossover	D. + C.	HMM	No
[8]	Serial and Parallel sampling	D. + C.	Any	No
[11]	Instantaneous PF	D. + C.	HMM	Yes
[12]	Alternating IS and WR inside time slice	D. + C.	Highly serial	Yes

subsets. Due to this distinction, in the experimental section, to show the performance of our framework, we select DBNs that show the advantages of our approach. As the experiments show there is an abundance of problems of this kind.

5 Experimental Results

The results presented in this paper compare our PFs with the KLPF because it is the generic algorithm for DBNs[1].

The first set of experiments, performed with simulated data compare the performance of the PFs using the Root Square Mean Error (RMSE) between the mean value of the particles estimates for each PF and the true value of the hidden variables at the last step of the simulation. In the second set, carried out with real world data, we use the different PFs to score the structure of a DBN given the measurements. In both cases, our PFs perform better than KLPF.

5.1 Simulated Experiments

This section compares our PFs, working with a limited number of particles, with KLPF in two complex real problems modeled by different DBNs. We have selected them because the structures of their DBNs, with multiple hidden variables that can be sampled in parallel, let them benefit significantly from our PFs. In fact, the second one is a simplification of our original problem: sea rescue where an Unmanned Air Vehicle (UAV) has to track several objects that are in the water and whose initial positions is not completely known. The first problem is a modification demonstrating our PFs on a more complex DBN. In both cases, we want to localize a set of O mobile objects given the information provided by the existing sensors. To be able to distinguish the variables related with the l-th object, in this section some variables names have a subindex (V_l).

In the first problem, each mobile object (M_l) is repelled by another (M_k) with a common unknown force (F), and the position of each mobile is observed by a different

[1] A preliminar analysis, excluded for space limitation, showed that the Serial PF by [10,12] worked even worst than KLPF due to the highly parallel structure of our DBNs.

sensor (S_l). The PFs have to estimate the probability of the hidden variables (M_l, F) given the measurements (S_l) for the DBN whose structure is defined by $\mathrm{Pa}_{t-1}(M_l) = \{M_l, M_{rem(l+1,O+1)}, F\}$, $\mathrm{Pa}_{t-1}(F) = \{F\}$ and $\mathrm{Pa}_t(S_l) = \{M_l\}$. The structure of this DBN let us divide the hidden variables in $O+1$ groups, each with a hidden variable and assign to the weights of each mixture the likelihoods of the measurement associated with each hidden variable ($\alpha_j^{\{F\}} = 1$ and $\alpha_j^{\{M_l\}} = \mathrm{p}\big(s_l|\mathrm{pa}_t^{(i)}(S_l)\big)$). We also make the proposal of each hidden variable equal to its prior transition model. Although each hidden variable belongs to a different group, the difficulty of this problem comes from the fact that they are highly coupled by the $\mathrm{Pa}_{t-1}(M_l)$ relation.

The second problem is an abstraction of a sea rescue problem with UAVs. The objects (M_l) move with the direction of the sea current and wind (E) and the sensors, which are onboard the UAV (U), are only able to detect (D_l) the mobiles that are within a circular area of the UAV and provide their position (S_l) when detected.The PFs have to estimate the probability of the hidden variables (M_l, E) given the observed ones (U, D_l, S_l) for the DBN whose structure is defined by $\mathrm{Pa}_{t-1}(M_l) = \{M_l, E\}$, $\mathrm{Pa}_{t-1}(E) = \{E\}$, $\mathrm{Pa}_t(D_l) = \{M_l, U\}$ and $\mathrm{Pa}_t(S_l) = \{M_l, D_l\}$. Again, the structure of this DBN let us divide the hidden variables in $O+1$ groups, each with a hidden variable and assign to the weights of each mixture the product of the likelihoods of the measurements associated with each hidden variable ($\alpha_j^{\{E\}} = 1$ and $\alpha_j^{\{M_l\}} = \mathrm{p}\big(d_l|\mathrm{pa}_t^{(i)}(D_l)\big)\mathrm{p}\big(s_l|\mathrm{pa}_t^{(i)}(S_l)\big)$). We also make the proposal of each M_l equal to its prior transition model. However, to let our PFs explore a bigger region of E, its proposal distribution is the prior transition model with extra noise. Although this problem is less coupled than the first one, it is also difficult due to the use of multiple measurements per hidden variable and the fact that when the object is not inside the circular area its position is not observed.

For each problem, we run to types of experiments. In both types, we measure at the last step of the simulation the Root Mean Square Error (RMSE) between the mean value of the particles estimates for each PF and the true value of the hidden variables.

In the first type, we calculate the RMSE over the same experiment 50 times for the KLPF, and our "trajectory" and "instantaneous" PFs, initializing the particles in each experiment and PF randomly using the same initial proposal. We then use the RMSE to clasify the experiments in "convergent" and "divergent" according with a selected threshold. Table 2a and 2b show the number of convergent runs and the mean over convergent runs of RMSE of the three PFs for each problem. Our PFs, which obtain similar solutions, hasn't been tested with more than 100 particles and the RMSE is not defined (n/d) when no run converges. For the first problem, Table 2a shows that while our PF convergence increases with N, no runs of KLPF converge. The bad results of KLPF are due to the high dependence of the hidden variables on each other, which is especially problematic for a global localization problem where the particles are created randomly in all the regions of the space. However, as our PFs sample independently from the more likely regions of each group of variables, the estimated positions of the mobiles can converge more easily to their real values. For the second problem, Table 2b shows that the convergence of all PFs increases with N, although our PFs have a higher velocity of growth. Hidden variables are not strongly interdependent and particles are initialized randomly in a smaller region of the space increasing chances of starting with

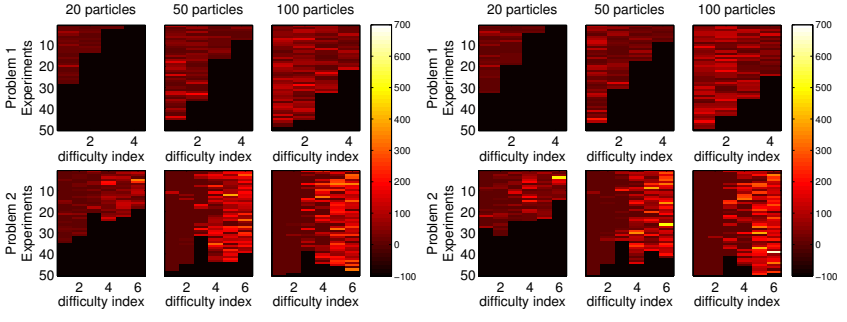
Table 2. First experiment: Number of convergent runs (conv.) and RMSE (mean over convergent runs) of the two PFs for experiments of problems 1 and 2 with L=10

(a) problem 1

# particles		20	50	100	500	1000
KLPF	conv.	0	0	0	0	0
	RMSE	n/d	n/d	n/d	n/d	n/d
"Trajectory" PF	conv.	7	30	42		
	RMSE	810	662	414		
"Instant." PF	conv.	12	27	36		
	RMSE	830	620	410		

(b) problem 2

# particles		20	50	100	500	1000
KLPF	conv.	4	7	8	22	23
	RMSE	832	1650	457	362	392
"Trajectory" PF	conv.	24	37	46		
	RMSE	892	483	382		
"Instant." PF	conv.	24	34	46		
	RMSE	956	429	382		



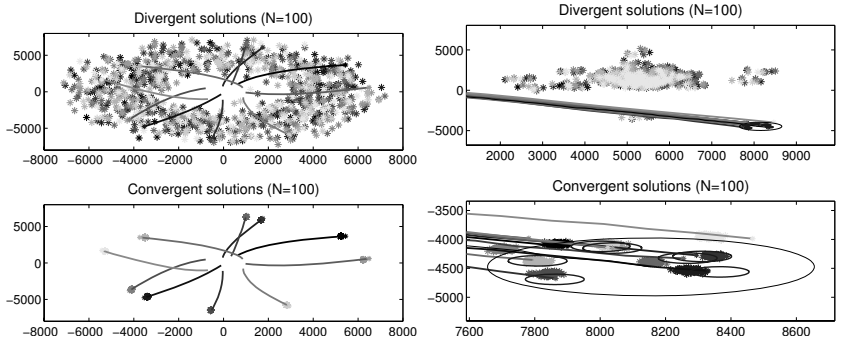
(a) "Trajectory" PF (b) "Instantaneous" PF

Fig. 3. Second experiment: Comparing KLPF with the "trajectory" and "instantaneous" PFs with the same initial particles

a good particle. This leads to improved performance of KLPF and superior performance of our PFs, both respect to the results observed in problem 1.

In the second type of experiments, we calculate the RMSE for the KLPF and our two PFs, with the same initial particles for all of them with the purpose of comparing how the filters work under the same initial conditions. The complete experiment consists on running the three filters for different number of particles (20, 50 and 100), with different initialization region proposals (4 for problem 1 and 6 for problem 2), and the same initial particles. For the convergent runs[2] of each of our PFs, we represent in Figure 3 the quotient of the RMSE of KLPF by the RMSE of one of our PFs ("Trajectory" in 3a and "Instantaneous" in 3b) for each problem (figure row), number of particles (figure column), initialization regions (x-axis, incrementing the size of the region, and so the difficulty of the problem with the number) and experiment number (y-axis). Quotients bigger than 1 show that our corresponding PF was better while smaller worst. The divergent runs are represented in black (set to -100). For problem 1, we can observe how incrementing the number of particles helps each of our PFs and how incrementing

[2] We use the same convergence criterium as in experiment 1.



(a) Problem 1 (b) Problem 2

Fig. 4. Particle positions at the final step (pluses) and real trajectory (lines) of the 10 mobile objects (each with a different grey level) of a convergent and divergent run of the PFs. The big circle in Problem 2 represents the area around the UAV while the little ones are centered around M_l for the detected mobiles.

the size of the region makes it worst. As none runs of the KLPF converge, each of our PFs outperforms the KLPF for all its convergent runs (quotient is always bigger than 1). For the second problem, when the difficulty index is small, KLFP and each of our PFS obtain similar RMSE, but as the difficulty index grows, the convergent runs of our corresponding PF get better than KLPF. Incrementing the number of particles also increments the number of convergent runs of our PFs. So, in short, our PFs deals better than KLPF with poorly initialized particles and complex problems.

Finally, Fig. 4 shows examples of convergent and divergent solutions for each of the problems. Fig. 4a, for problem 1, shows the position of the mobiles in the particles at the final iteration (pluses) and the real trajectory (lines) for a typical run of KLPF (that always diverges) and of our PFs. Fig. 4b, for problem 2, represents the position of the mobiles in the particles at the final iteration (pluses) and the real trajectory (lines); the circular area around the UAV (big circle) and an area around the measured position $s_{l,t}$ of the detected mobiles (small circles). Note the different axes used in each case, due to the wider distribution of the particles for the divergent example.

5.2 Real Data

For the real data experiment we have used a functional Magnetic Resonance Imaging (fMRI) dataset [14]. FMRI signal represents the Blood Oxygenation Level Dependent (BOLD) response measured in a small cubic region of the brain (voxel). BOLD response is itself governed by the underlying hidden neural activity. As the model of the BOLD signal generation from neural activity we have used the hemodynamic forward model based on a coupled system of ordinary differential equations [15].

The number of voxel measurements collected per time point in a typical fMRI experiment is too large to model directly. Thus voxel time courses were averaged on a per Region of Interest (ROI) basis. ROIs were selected according to the widely used Talairach database [16].

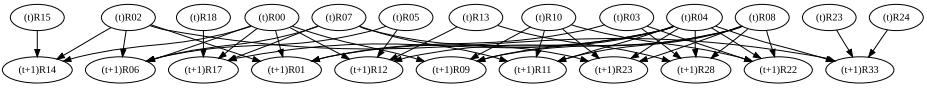
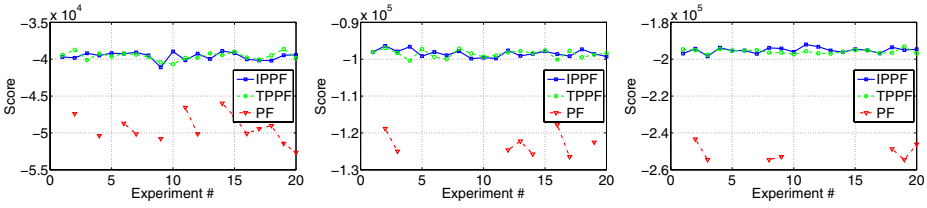


Fig. 5. The network for a subset of regions of interest (ROIs) from the Talairach anatomical atlas database. ROI names are replaced by short labels since they are not significant fr our work. Note that each ROI is observed through its indirect measurement by fMRI.



(a) 20 particles (b) 50 particles (c) 100 particles

Fig. 6. Cross validation log likelihood score plots for the instantaneous (IPPF) and trajectory (TPPF) parallel PFs, as well as regular particle filter (PF). Not displayed points mean that kernel density estimator did not produce a valid result.

In order to obtain the underlying structure of the DBN, we have used the approach that treats fMRI data as fully observed and quantizes it into categorical representation [14]. Among the discovered DBN families we have used several most significant ones according to the cross validation procedure (t-test with p-value of 0.05). This resulted in a DBN of 42 hidden variables per slice. Figure 5 shows a small portion of the hidden structure of the DBN we use. The observation nodes that are present for each ROI in a time slice are not shown.

The two novel parallel PF algorithms as well as the KLPF were run on this dataset for 50 time points (100 seconds with 2 seconds fMRI sampling rate). Since the ground truth for neural activity of ROIs is unknown in this dataset, for evaluation we have used cross validation log likelihood based on kernel density estimators with Gaussian kernels and automatically chosen variance value using a cross validation procedure [17].

The results for 20 runs using 20, 50 and 100 particles are shown in Figure 6. Parallel PFs show higher score (better) although are not much different in their performance. KLPF has considerably lower mean score as well as a higher variance. Note how the score decreases as the number of particles grows. This is an expected behavior with Parzen window estimators, that is due to the smoother and generally wider estimates in the cases of more data that lead to lower probability values. Thus in general this result supports the one obtain on simulated data.

6 Conclusions

This paper presents two PFs to estimate the trajectory or current state of the hidden variables of a DBN. It consists of a highly configurable proposal that samples in parallel the values of different subsets of hidden variables to build a whole particle whose weight is updated according to the sampling proposal and the estimation problem.

Our PFs explore different subspaces of the state space in parallel while performing the sampling step, and the whole space as a block while updating the weights. Inside each subspace the exploration is carried out by means of a mixture distribution, whose weights and components are selected by the user to control the regions of the space it wants to explore. For the whole space step, either the "trajectory" update weight function by Koller and Lenser [9] or the "instantaneous" update proposed in this paper can be used. In our experiments, our two filters show similar performances.

Our tests show that our PFs usually better sample the state space when the particles are initialized in spread regions, hence it is superior to KLPF in keeping track of the hidden state while needing smaller number of particles.

Finally, in combination with the Serial PF method proposed in [12], we can build a Parallel & Serial PF that updates the weights according to the sampling functions.

Acknowledgments

This work was supported by the Spanish Grants DPI2006-15661-C02-01 and CAM S-0505/DPI 0391. Further, Dr. Besada-Portas was supported by the Spanish post-doctoral Grant EX-2007-0915, Dr. Plis by NIMH Grant 1 R01 MH076282-01, and Dr. Lane by NSF Grant IIS-0705681. The authors also thank the Aula Sun-UCM for providing access to their computational resources for doing parts of the experiments.

References

1. Doucet, A., Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo methods in practice. Springer, Heidelberg (2001)
2. MacCormick, J., Isard, M.: Partitioned sampling, articulated objects, and interface-quality hand tracking. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 3–19. Springer, Heidelberg (2000)
3. Doucet, A., de Freitas, N., Murphy, K., Russell, S.: Rao-blackwellised particle filtering for dynamic bayesian networks. In: 16th Conference on Uncertainty in Artificial Intelligence, pp. 176–183 (2000)
4. Vaswani, N.: Particle filters for infinite (or large) dimensional state spaces-part 2. In: IEEE ICASSP (2006)
5. Ng, B., Peshkin, L.: Factored particles for scalable monitoring. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, pp. 370–377. Morgan Kaufmann, San Francisco (2002)
6. Das, S., Lawless, D., Ng, B., Pfeffer, A.: Factored particle filtering for data fusion and situation assessments in urban environments. In: 8th International Conference on Information Fusion (July 2005)
7. Park, S., Hwang, J., Rou, K., Kim, E.: A new particle filter inspired by biological evolution: Genetic Filter. Proceeding of World Academy of Science, Engineering and Technology 21, 57–71 (2007)
8. Brandao, B.C., Wainer, J., Goldenstein, S.K.: Subspace hierarchical particle filter. In: XIX Brazilian Symposium on Computer Graphics and Image Processing (2006)
9. Koller, D., Lerner, U.: Sampling in factored dynamic systems. In: Sequential Monte Carlo in Practice, pp. 445–464 (2001)

10. Maccormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. Intenational Journal of Computer Vision 39(1), 57–71 (2000)
11. Klaas, M., de Freitas, N., Doucet, A.: Toward practical n^2 Monte Carlo: the Marginal Particle Filter. In: Proceedings of UAI 2005, Arlington, Virginia, pp. 308–331. AUAI Press (2005)
12. Rose, C., Saboune, J., Charpillet, F.: Reducing particle filtering complexity for 3D motion capture using dynamic bayesian networks. In: 23th AAAI Conference on Artificial Intelligence (2008)
13. Pitt, M.K., Shephard, N.: Filtering via simulation: Auxiliary Particle Filters. Journal of the American Statistical Association 94(446), 590–599 (1999)
14. Burge, J., Lane, T., Link, H., Qiu, S., Clark, V.P.P.: Discrete dynamic bayesian network analysis of fMRI data. Human Brain Mapping (November 2007)
15. Friston, K.J., Harrison, L., Penny, W.: Dynamic Causal Modelling. NeuroImage 19(4), 1273–1302 (2003)
16. Lancaster, J.L., Woldorff, M.G., Parsons, L.M., Liotti, M., Freitas, C.S., Rainey, L., Kochunov, P.V., Nickerson, D., Mikiten, S.A., Fox, P.T.: Automated Talairach atlas labels for functional brain mapping. Human Brain Mapping 10(3), 120–131 (2000)
17. Hofmann, R., Tresp, V.: Discovering structure in continuous variables using bayesian networks. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) Advances in Neural Information Processing Systems, vol. 8, pp. 500–506. MIT Press, Cambridge (1996)