# Sequential Auxiliary Particle Belief Propagation

Mark Briers[1,2], Arnaud Doucet[3] and Sumeetpal S Singh[1]

[1] Cambridge University Engineering Department, Cambridge, CB2 1PZ, UK
[2] QinetiQ Ltd, Malvern, WR14 3PS, UK
[3] University of British Columbia, Vancouver, V6T 122, Canada
e-mail: mb511,sss40@eng.cam.ac.uk, arnaud@cs.ubc.ca

**Abstract** – *This paper discloses a novel algorithm for efficient inference in undirected graphical models using Sequential Monte Carlo (SMC) based numerical approximation techniques. The methodology developed, titled "Auxiliary Particle Belief Propagation", extends the applicability of the much celebrated (Loopy) Belief Propagation (LBP) algorithm to non-linear, non-Gaussian models, whilst retaining a computational cost that is linear in the number of sample points (or particles). Furthermore, we provide an additional extension to this technique by analysing temporally evolving graphical models, a problem which remains largely unexplored in the scientific literature. The work presented is thus a general framework that can be applied to a plethora of novel distributed fusion problems. In this paper, we apply our inference algorithm to the (sequential problem of) articulated object tracking.*

**Keywords:** Belief propagation, particle filter, Monte Carlo, sequential inference, graphical models.

## 1   Introduction

Graphical models form an intuitively appealing marriage of probability theory and graph theory. Their use promotes interaction between statisticians and relevant problem-domain experts, a point which is prominent in Bayesian analysis. The graphical representation of the links between random variables has allowed the development of a plethora of inference mechanisms, most of which exploit structure within the model to promote efficient inference. The much-celebrated Belief Propagation (BP) algorithm[7] is a particular example of such a technique, providing an exact solution in cases where one can perform analytic calculations in tree structured graphical models. Moreover, an approximate algorithm known as Loopy Belief Propagation (which is similar in its implementation as standard BP) has been employed on analytically tractable generic graphical models (i.e. those with "loops"), with varying degrees of success. The use of BP is, however, restricted to "static" inference problems, not least because inference with the existence of loops is an extremely difficult problem and is still to be fully understood.

This paper discloses an inference mechanism that combines BP and importance sampling (or particle filtering), the first step towards a generic framework for performing truely distributed fusion. Existing technologies[3, 4, 10] that have combined particle methods with BP have a computation that is quadratic in the number of particles, and tend to employ Gibbs sampling, which has to be executed until convergence. Our complementary technique, which we call Auxiliary Particle Belief Propagation (APBP), reduces the computation of existing particle-based techniques to being linear in the number of particles. In addition to this, we provide a *novel* extension which allows one to perform *online* inference in temporally evolving graphical models (in a distributed fashion) using such numerical (particle based) approximations, a problem which has been largely overlooked in the literature (thus making comparisons infeasible). As stated, the development of this framework is the first stage in allowing the future development of a truly distributed tracking architecture using particle methods.

The paper is therefore organised as follows: in section 2 we provide a brief overview of inference in graphical models, specifically based upon the BP algorithm. Then, in section 3, we discuss an empirical approximation within the BP algorithm and highlight the computational problems associated with such an approach. In the following two sections, we disclose a novel efficient methodology that overcomes these computational restrictions and then extend this technique to the case of temporally evolving (undirected) graphical models in section 6. We demonstrate the success of this algorithm on the (sequential problem of) articulated object tracking in section 7, and finally conclude and discuss avenues of further work in section 8.

## 2   Inference in graphical models

### 2.1   Graphical model notation

In this section, we provide a very brief overview of (undirected) graphical models and refer the reader to the excellent reference [5] for a complete exposition. Note that directed graphical models (informally referred to as Bayesian networks) can be converted to undirected graphical models[12].

The joint probability for the latent $(x)$ and observed $(y)$ variables in a general graphical model can be written as follows:

$$p(x,y) \propto \prod_{i \in \mathcal{V}} p_i(y_i|x_i) \prod_{c \in \mathcal{C}} \psi_c(x_c),$$

where $\mathcal{V}$ denotes the set of vertices (random variable indices) and $\mathcal{C}$ denotes the set of cliques (fully connected subsets of nodes). In this model, each latent variable has an

associated observed variable. To simplify the presentation, we restrict ourselves to pairwise Markov potentials:

$$p(x, y) \propto \prod_{i \in \mathcal{V}} p_i(y_i|x_i) \prod_{(j,i) \in \mathcal{E}} \psi_{j,i}(x_i, x_j),$$

where $\mathcal{E}$ denotes the set of edges joining two (potentially clustered) nodes. The pairwise Markov potential model is formed by using graph theoretic clustering algorithms over the cliques. Additionally, these potentials are non-negative functions describing the stochastic relationship between two neighbouring random variables and satisfy the integrability assumption:

$$\int \psi_{j,i}(x_i, x_j)dx_i < \infty, \quad \int \psi_{j,i}(x_i, x_j)dx_j < \infty.$$

## 2.2 Belief Propagation

In this paper, and in most distributed fusion applications, we are concerned with determining the marginal posterior distribution $p(x_i|y)$. That is, we are interested in the following:

$$p(x_i|y) \propto \int \cdots \int p(x, y)dx_{\backslash\{i\}}$$
$$\propto \int \cdots \int \prod_{i \in \mathcal{V}} p_i(y_i|x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{i,j}(x_i, x_j)dx_{\backslash\{i\}},$$

where $x_{\backslash\{i\}} \triangleq \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_K\}$.

Now, one can reorder this integral to exploit the structural properties of the graphical model using the elimination algorithm to reduce the computational effort. However, determining such an ordering in the most general case whilst trying to minimise the computations (i.e., to exploit the redundancy in the structure) is an NP-hard problem. The (Loopy) Belief Propagation algorithm[7] is an approximation that imposes the ordering:

$$p^n(x_i|y) \propto p_i(y_i|x_i) \prod_{j \in \mathcal{N}(i)} m_{j,i}^n(x_i) \qquad (1)$$

where $\mathcal{N}(i)$ defines the neighbourhood of node $i$, and the superscript $n$ denotes the iteration number. The values $m_{j,i}(x_i)$ are generally "intuitively" stated as being *messages* from node $j$ to node $i$, and are defined as follows:

$$m_{j,i}^n(x_i) \propto \int \psi_{j,i}(x_i, x_j)p_j(y_j|x_j)$$
$$\times \prod_{k \in \mathcal{N}(j)\backslash i} m_{k,j}^{n-1}(x_j)dx_j. \qquad (2)$$

This recursion is constructed through dynamic programming. Clearly, these calculations are tractable for Gaussian and finite state space models (with the traditional Kalman smoother being a special case of BP). In the case of tree-structured graphical models, the algorithm is guaranteed to converge towards $p(x_i|y)$ (with aforementioned tractable calculations). In graphical models with loops, convergence is not guaranteed but good empirical performance has been

shown in the literature[6]. Some theoretical properties of Loopy BP (LBP) have been investigated in reference [12].

Our aim is to remove all assumptions of analytic tractability of the integrals involved and to construct a particle approximation of the messages and beliefs.

# 3 Towards particle belief propagation

As discussed, the belief propagation procedure is decomposed into two stages: message update (2) followed by posterior belief update (1). We would like to use an importance sampling approximation of these probability measures. The optimal importance function, i.e., optimal *with respect to minimising the variance of the importance weights*, is now discussed for each stage[1].

## 3.1 Optimal importance function

Assume for now that, at iteration $n-1$, we have a pointwise approximation of the message (2) from node $j$ to node $i$ (for all $(i, j) \in \mathcal{E}$), given as[2]:

$$\hat{m}_{j,i}^{n-1}(x_i) = \sum_{l=1}^N w_{(j,i)}^{(l),n-1} \psi_{j,i}(x_i, x_{(j)}^{(l),n-1}),$$

where $w_{(j,i)}^{(l),n-1}$ is the (normalised) $l^{th}$ weight for the message approximation of $j$ to $i$ at iteration $n-1$, and $x_{(j)}^{(l),n-1}$ is the $l^{th}$ particle at node $j$ at iteration $n-1$ for this message. The optimal choice of importance function for the approximation of (2) is given as:

$$\tilde{q}_{j,i}^{opt}(x_j|y_j) \triangleq \frac{p_j(y_j|x_j) \prod_{k \in \mathcal{N}(j)\backslash i} \hat{m}_{k,j}^{n-1}(x_j)}{\int p_j(y_j|x_j) \prod_{k \in \mathcal{N}(j)\backslash i} \hat{m}_{k,j}^{n-1}(x_j)dx_j}.$$

For the belief update (1) at iteration n, assuming a pointwise approximation of the messages from each of the neighbouring nodes, the optimal importance function is:

$$\tilde{q}_i^{opt}(x_i|y_i) \triangleq \frac{p_i(y_i|x_i) \prod_{j \in \mathcal{N}(i)} \hat{m}_{j,i}^n(x_i)}{\int p_i(y_i|x_i) \prod_{j \in \mathcal{N}(i)} \hat{m}_{j,i}^n(x_i)dx_i}.$$

As one can clearly see, the form of the importance functions are structurally identical for both the message and belief update stages (that is, they are both products of mixtures of similar terms), and as such, one can devise a common methodology from which to sample. Before doing so, it is worth commenting on the computational and tractability restrictions of such a choice of importance function.

Since both of the proposal distributions involve a product of mixtures of terms, then for a large number of particles $N$ and a dense graphical model (that is, the cardinality of the set of terms in the product (denoted $K$) is large) we have an explosion in the computational complexity (or number

---

[1]Henceforth, optimal importance function will always mean optimal with respect to minimising the variance of the importance weights.

[2]Details of how to obtain such an approximation are given later in the paper.

of mixture components) in the proposal distribution. Moreover, calculation of the normalising constant in each of the importance functions is intractable (if we could calculate this integral, then it is likely that we would not have to make an approximation!). To circumvent the computational problem we devise a novel[3] strategy that introduces an auxiliary variable in analogy to the Auxiliary Particle Filter (APF)[8]. The auxiliary variable clearly increases the dimensionality of the sampling space under consideration since we are working on the product space, but allows for a computationally efficient algorithm. To circumvent the tractability problems, one has to use an approximation to the importance function used (on the extended space), which will be detailed later in the paper. We now generalise the discussion to sampling from a product of mixtures using importance sampling before returning to the auxiliary particle belief propagation technique.

## 4   Generic concept: sampling from a product of mixtures

The general aim is to sample from the following:

$$p(x) = \frac{\phi(x)\prod_{i=1}^{K} f_i(x)}{\int \phi(x)\prod_{i=1}^{K} f_i(x)dx}, \qquad (3)$$

where

$$f_i(x) = \sum_{l=1}^{N} f_i^{(l)}(x)$$

is a mixture comprised of $N$ terms, $x \in \Re^d$ for each $i$, and $\phi(x)$ is the likelihood term.

### 4.1   Auxiliary variable

As stated, we consider the introduction of a latent variable for each term in the product $i$:

$$\theta_i \in \{1, 2, \ldots, N\}, \qquad i = 1, \ldots, K,$$

with $\theta_{1:K} = \{\theta_1, \ldots, \theta_K\}$. The discrete valued random variable $\theta_i = m_i$ denotes the mixture component in $f_i$ from which the sample is drawn. We may write:

$$p(x) = \sum_{\theta_{1:K}} p(x, \theta_{1:K})$$
$$= \sum_{\theta_{1:K}} p(\theta_{1:K})p(x|\theta_{1:K}).$$

The terms in this factorisation of the joint distribution are given as:

$$p(\theta_{1:K} = m_{1:K}) = Z^{-1}\int \phi(x)\prod_{i=1}^{K} f_i^{(m_i)}(x)dx, \quad (4)$$

---

[3]It is worth commenting that reference [3] described an importance sampling perspective but omitted the actual importance function used within the paper and did not consider the computational aspects that are fundamentally important within such an approach (and so did not use an auxiliary variable as will be described here).

where $Z^{-1}$ denotes a normalising constant, and:

$$p(x|\theta_{1:k} = m_{1:K}) = \frac{\phi(x)\prod_{i=1}^{K} f_i^{(m_i)}(x)}{\int \phi(x)\prod_{i=1}^{K} f_i^{(m_i)}(x)dx}. \quad (5)$$

Now, the procedure to sample such a product of mixtures is as follows:

- sample the latent variables $\theta_{1:K} \in (\{1, 2, \ldots, N\}^K)$ according to the probability given in (4);

- Having sampled $\theta_{1:K}$, we then sample $X$ from the product of densities in (5). It is easy to show that $X$ is distributed according to the product of mixtures (3).

We will return to mechanics of how to actually draw samples from (5) later in the paper. For now, we assume that for any $(m_1, \ldots, m_K) \in (\{1, 2, \ldots, N\}^K)$, one can sample from the product $\phi(x)\left(\prod_{i=1}^{K} f_i^{(\theta_i)}(x)\right)$, and that the integral over this term w.r.t $x$ is analytically tractable.

The actual difficulty lies in generating the latent variable $\theta_{1:K}$ as there are $N^K$ possible configurations. Enumerating all such possible configurations would be wasteful since one might as well just sample each component! An approximation is therefore needed. One approach is to use the Gibbs sampler, which is similar in principal to the approach adopted in references [4, 10]. The marginal distribution of $\theta_i$ conditioned on $\theta_{\backslash\{i\}} \triangleq \{\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_K\}$ is

$$\Pr\left(\theta_i \middle| \theta_{\backslash\{i\}} = m_{\backslash\{i\}}\right) \propto \int \phi(x)f_i^{(\theta_i)}(x)\prod_{\substack{j=1\\j\neq i}}^{K} f_j^{(m_j)}(x)dx,$$

with $\theta_{\backslash\{i\}}$ and $m_{\backslash\{i\}}$ being deemed obvious notation from the context. Assuming we run $\kappa$ Gibbs sampling iterations, the computational cost of existing Non-Parametric Belief Propagation algorithms is $O(\kappa K N^2)$. Moreover, MCMC within a particle filtering context has been used [2] but is sometimes referred to as being inefficient[8]. To avoid such a prohibitive computational cost (which is particularly important when one considers a temporally evolving system) and a potentially slowly mixing Markov chain, we adopt an importance sampling approach which reduces the computation to $O(NK)$ computations, that are able to be parallelised.

### 4.2   Auxiliary variable: an importance sampling perspective

Since the calculation of the (true) optimal proposal is computationally intractable then we have to make an approximation, that is, we introduce an importance function. We propose two such choices of importance function, although it is anticipated that there may be other approximations that may improve the performance of our algorithm.

#### 4.2.1   Case 1: Independent component labels

The simplest form of proposal distribution for the joint auxiliary variable is through assuming independence between

each dimension (terms in the product), such that we have:

$$q(\theta_{1:K}) = \prod_{i=1}^{K} q_i(\theta_i), \qquad (6)$$

where $q_i(\cdot)$ is a pmf on $1, \ldots, N$. A sensible choice would be

$$q_i(\theta_i = m_i) \propto \int \phi(x) f_i^{(m_i)}(x) dx. \qquad (7)$$

Clearly this cannot be used when one cannot calculate the above integral, and so further approximations have to made as detailed later in the paper. Note that the computational cost of this approach is $O(KN)$.

### 4.2.2 Case 2: Conditionally dependent labels

Instead of complete independence of the importance labels, we can factorise the joint distribution as follows:

$$q(\theta_{1:K}) = \prod_{i=1}^{K} q_{i|1:i-1}(\theta_i|\theta_{1:i-1}) \qquad (8)$$

where we define $q_{1|0}(\cdot) \triangleq q_1(\cdot)$, as given in equation (7). The success of this choice of importance function (in terms of minimising the variance of the importance weights) is clearly dependent upon the ordering of the terms in the latent space, but we assert that by relaxing the assumption (6) one should see an improvement in performance.

Now, a sensible choice of importance function for $i \geq 1$, $q_{i|1:i-1}(\cdot | m_1, \ldots, m_{i-1})$, a pmf on $\{1, 2, \ldots, N\}$ for each $(m_1, \ldots, m_{i-1}) \in \{1, 2, \ldots, N\}^{i-1}$ is given by:

$$q_{i|1:i-1}(\theta_i| \theta_{1:i-1} = m_{1:i-1}) \propto \int \phi(x) f_i^{(\theta_i)} \prod_{j=1}^{i-1} f_j^{(m_j)} dx.$$

It is worth noting, however, that the use of this proposal is computationally more restrictive than (6), (it is order $O(KN^2)$) and suffers from the same analytic tractability issues.

## 5 Auxiliary particle belief propagation

It is instructive at this point to remove oneself from the abstract concepts used in the previous section and return to the numerical approximation of the posterior belief (and the intermediate messages) in a (time invariant) graphical model. The problem that we face is that it is not possible in the most general case to sample from the product appearing in the proposal distributions, nor are we able to analytically calculate the integrals appearing in the auxiliary variable proposal. An approximation to the optimal importance function therefore has to be made. In this paper we consider an Unscented approximation, in analogy to the Unscented Particle Filter[11], since such an approximation is deemed to be most flexible. Other similar approximations, such as a first order Taylor series approximation or a variational approximation, could also be used. As was highlighted earlier in the paper, the proposal distributions for the message and belief updates are structurally very similar and so we do not

differentiate between the two in the ensuing discussion, for brevity.

We stress at this point that the approximations introduced in the previous section are approximations of the importance functions used; no such *ad hoc* approximations are used in the approximation of the marginal posterior probabilities nor the messages. Such entities are approximated using the theoretically rigorous (sequential) importance sampling mechanism, a concept which is known to be asymptotically optimal. Whilst it may appear that many approximations have been suggested in the previous section, this is clearly not the case for the quantities of interest.

### 5.1 Sampling the component labels

To sample the component labels in the product of mixtures, one is required to calculate the following integral (we consider (6) since an analogy to the more complicated case will be discussed in the following section):

$$q_i(\theta_i = m_i) \propto \int p_j(y_j|x_j)\psi_{j,i}(x_i, x_j^{m_i}) dx_i$$

where for clarity we have dropped the dependence on the iteration number. In the most general this integral is analytically intractable and so an approximation has to be made, although this does not compromise any theoretical considerations since this is an approximation only to the importance function. An Unscented approximation to this integral, as suggested in reference [1], would involve producing a set of sigma points $\{\widehat{x}_j^{(l)}\}_{l=1}^{W}$ with corresponding weights $\alpha^{(l)}$ in the usual manner, to allow one to determine the mean:

$$q_i(\theta_i = m_i) \propto \sum_{l=1}^{W} \alpha^{(l)} p_j\left(y_j|\psi_{j,i}(x_i^{(m_i)}, \widehat{x}_j^{(l)})\right).$$

### 5.2 Sampling the state

The Unscented approximation for the proposal distribution is conducted in two stages: we first approximate the product term (denoted the "prior") and then approximate the Kalman filter update by incorporating information from the "likelihood". So, for the first stage, we form an Gaussian approximation to each of the terms in the product using a set of weighted sigma points. The mean and covariance are calculated for all terms in the product ($k \in \mathcal{S}$) as:

$$\mu_x^{j,k} = \sum_{l=1}^{W} \alpha^{(l)} \psi_{j,k}(x_k^{(m_k)}, \widehat{x}_j^{(l)})$$

$$P_{xx}^{j,k} = \sum_{l=1}^{W} \alpha^{(l)} (\psi_{j,k}(x_k^{(m_k)}, \widehat{x}_j^{(l)}) - \mu_x^{j,k})$$
$$\times (\psi_{j,k}(x_k^{(m_k)}, \widehat{x}_j^{(l)}) - \mu_x^{j,k})^T.$$

The mean and covariance of the product of terms are calculated:

$$\mu_x^j = P_{xx}^j \left(\sum_{k \in \mathcal{S}} (P_{xx}^{j,k})^{-1} \mu_x^{j,k}\right)$$

$$P_{xx}^j = \left(\sum_{k \in \mathcal{S}} (P_{xx}^{j,k})^{-1}\right)^{-1},$$

with a further set of sigma points chosen from the Gaussian:

$$N\left(\left[\begin{array}{c} \mu_x^j \\ 0 \end{array}\right], \left[\begin{array}{cc} P_{xx}^j & 0 \\ 0 & R \end{array}\right]\right),$$

where $R$ is the measurement noise covariance. The sigma points are then used to determine the following moments:

$$\mu_y^j = \sum_{l=1}^{W} \tilde{\alpha}^{(l)} \phi_j(\hat{x}_j^{(l)}, \hat{\nu}_j^{(l)})$$

$$P_{yy}^j = \sum_{l=1}^{W} \tilde{\alpha}^{(l)} \phi_j(\hat{x}_j^{(l)}, \hat{\nu}_j^{(l)}) - \mu_y^j)$$

$$\times (\phi_j(\hat{x}_j^{(l)}, \hat{\nu}_j^{(l)}) - \mu_y^j)^T$$

$$P_{xy}^j = \sum_{l=1}^{W} \tilde{\alpha}^{(l)} (\hat{x}_j^{(l)} - \mu_x^j)(\phi_j(\hat{x}_j^{(l)}, \hat{\nu}_j^{(l)}) - \mu_y^j)^T,$$

where $\phi_j$ is the usual functional representation of the measurement to the latent state and measurement noise, $\nu_j$. Finally, the mean $\mu^j$ and covariance $P^j$ of the Unscented approximation of the optimal importance function are calculated as follows:

$$\mu^j = \mu_x^j + P_{xy}^j (P_{yy}^j)^{-1}(y_j - \mu_y^j) \qquad (9)$$

$$P^j = P_{xx}^j - P_{xy}^j (P_{yy}^j)^{-1} P_{xy}^T. \qquad (10)$$

### 5.3 Weight update

The normalised weight for the belief update can now be written as:

$$w_i^{(l),n} \propto w_i^{(l),n-1} \times$$

$$\frac{p_i(y_i|x_i^{(l),n}) \prod_{j \in \mathcal{N}(i)} w_{(j,i)}^{(\theta_j^{(l)}),n} \psi_{j,i}(x_i^{(l),n}, x_{(j)}^{(\theta_j^{(l)}),n})}{q(\theta_{1:K}^{(l)}) q(x_i^{(l),n}|\theta_{1:K}^{(l)})}, \qquad (11)$$

and the normalised weight for the message update can be written as:

$$w_{(j,i)}^{(l),n} \propto w_{(j,i)}^{(l),n-1} \times$$

$$\frac{p_j(y_j|x_{(j)}^{(l),n}) \prod_{k \in \mathcal{N}(j) \setminus i} w_{(k,j)}^{(\theta_k^{(l)}),n-1} \psi_{k,j}(x_{(j)}^{(l),n}, x_{(k)}^{(\theta_k^{(l)}),n-1})}{q(\theta_{1:K}^{(l)}) q(x_j^{(l),n}|\theta_{1:K}^{(l)})}. \qquad (12)$$

To ensure that the algorithm has converged, we employ a similar tactic to that in standard (discrete/Gaussian) cases, with the main difference being the introduction of a set of test functions that are integrated using the particle approximation. Consider the set of (user-defined) test functions: $\varphi_1, \ldots, \varphi_q$. Using the particle approximations of the beliefs at times $n$ and $n-1$, we can introduce the following rule of (algorithmic) convergence assessment:

$$\text{Terminate if: } \max_\alpha \left| < b_i^{(n)}, \varphi_\alpha > - < b_i^{(n-1)}, \varphi_\alpha > \right| < \epsilon,$$

for some suitable threshold $\epsilon > 0$, tested $\forall i \in \mathcal{V}$.

Pseudo-code for the APBP algorithm is given below:

---

AUXILIARY PARTICLE BELIEF PROPAGATION
For each node; While not converged:

1. For $i = 1, \ldots, N$ (message or belief update):

   - Sample the component label:
     - Using an unscented approximation to (6) or (8) as detailed in section 5.1, sample the component label.
   - Sample the state:
     - Conditioned on the component label, use an unscented approximation to the posterior as detailed in section 5.2 to form an importance function and sample the state.
   - Re-weight according to (11) or (12).

2. Resample using standard techniques if necessary.

---

## 6 Sequential auxiliary particle belief propagation

Perhaps the most important contribution within this paper is within this section; developing the ability to perform online efficient inference in a temporally evolving graphical model using Sequential Monte Carlo (SMC) techniques. Clearly, one could employ a standard particle filter on the joint space of all of the nodes in the graphical model and integrate to determine the marginal estimates, but this would be wasteful since one is not exploiting the conditional independence of the graphical structure. Moreover, such sampling would remove the ability to perform decentralised processing, which is a necessity in distributed fusion as well as other real world systems. We now extend our proposed APBP algorithm to account for a time dimension in the inference problem.

### 6.1 Sequential Belief Propagation

Let $x_{i,t}$ and $y_{i,t}$ denote the latent and observed variables at time $t$, respectively. We make the following standard Markov assumption for the time dimension:

$$p(x_t|x_{t-1}) = \prod_{i \in \mathcal{V}} p(x_{i,t}|x_{i,t-1}).$$

The marginal belief at time $t$ is then:

$$p(x_{i,t}|y_{1:t}) \propto p(x_{i,t}, y_t|y_{1:t-1})$$

$$\propto \int \prod_{k \in \mathcal{V}} p_i(y_{k,t}|x_{k,t}) p(x_{k,t}|y_{1:t-1})$$

$$\times \prod_{(i,j) \in \mathcal{E}} \psi_{j,i}(x_{i,t}, x_{j,t}) dx_{\setminus\{i\},t},$$

where we have assumed a time-inhomogeneous likelihood for notational simplicity. The belief update and message update, in analogy to (1) and (2) respectively, can be written as:

$$p^n(x_{i,t}|y_{1:t}) \propto p_i(y_{i,t}|x_{i,t})p(x_{i,t}|y_{1:t-1}) \prod_{j\in\mathcal{N}(i)} m^n_{j,i,t}(x_{i,t})$$

$$m^n_{j,i,t}(x_i) \propto \int \psi_{j,i}(x_{i,t},x_{j,t})p_j(y_{j,t}|x_{j,t})p(x_{j,t}|y_{1:t-1})$$
$$\times \prod_{k\in\mathcal{N}(j)\setminus i} m^{n-1}_{k,j,t}(x_{j,t})dx_{j,t},$$

where:

$$p(x_{j,t}|y_{1:t-1}) = \int p(x_{j,t}|x_{j,t-1})p(x_{j,t-1}|y_{1:t-1})dx_{t-1}.$$

It is noted that a similar formulation was provided in reference [3], where the authors (in the sequential case) use the (time-dependent) prior $p(x_{i,t}|x_{i,t-1})$ as their importance function. However, it is not clear how the authors circumvent the computational explosion/tractability issues present in such importance sampling schemes.

## 6.2  Weight update

The algorithm described within this section is a straightforward extension of the auxiliary concepts: one again samples an auxiliary variable (set of components in the products of mixtures) and then samples the state conditional on the auxiliary variable. The inclusion of the prior only brings about a further auxiliary variable (denoted $\gamma_j$ in the product, and thus increases the dimension of the auxiliary variable), and since the true distribution has changed, the normalised weight for the belief update is given as:

$$w^{(l),n}_{j,t} \propto w^{(l),n-1}_{j,t}p_j(y_{j,t}|x^{(l),n}_{j,t})\times$$

$$\frac{w^{(\gamma^{(l)}_j)}_{j,t-1}p(x^{(l),n}_{j,t}|x^{(\gamma^{(l)}_j)}_{j,t-1})\prod_{j\in\mathcal{N}(l)}w^{(\theta^{(l)}_i)}_{(j,i),t}\psi_{j,i}(x^{(\theta^{(l)}_i)}_{(i),t},x^{(l),n}_{j,t})}{q(\theta^{(l)}_{1:K},\gamma^{(l)}_j)q(x^{(l),n}_{j,t}|\theta^{(l)}_{1:K},\gamma^{(l)}_j)},$$

with the normalised weight for the message update given as:

$$w^{(l),n}_{(j,i),t} \propto w^{(l),n-1}_{(j,i),t}p_j(y_{j,t}|x^{(l),n}_{(j),t})w^{(\gamma^{(l)}_j)}_{j,t-1}p(x^{(l),n}_{(j),t}|x^{(\gamma^{(l)}_j)}_{j,t-1})\times$$

$$\frac{\prod_{k\in\mathcal{N}(j)\setminus i}w^{(\theta^{(l)}_k)}_{(k,j)}\psi_{k,j}(x^{(l),n}_{(j),t},x^{(\theta^{(l)}_l),n-1}_{(k),t})}{q(\theta^{(l)}_{1:K,t},\gamma^{(l)}_j)q(x^{(l),n}_{(j),t}|\theta^{(l)}_{1:K,t},\gamma^{(l)}_j,y_{1:t-1},y_{j,t})}.$$

Clearly one can perform "standard" particle filtering to determine an initial message for each of the nodes, by considering a particle approximation to the local posterior distribution $p(x_{t,i}|y_{1:t-1},y_{t,i})$. This allows for a more intuitive understanding for the initial messages and experimental results have shown that it has reduced the variance of the weights of the message passing algorithm.

## 7  Results

In this section, we demonstrate the SAPBP algorithm on articulated object tracking, specifically pose estimation of a human. It is not the intention of this section to perform novel "computer vision", we merely use this as a simple mechanism to display results from our algorithm.

### 7.1  Articulated object tracking

Articulated object tracking is becoming a pertinent problem in many application domains (from military to medical). In this paper, we consider the application of the SAPBP methodology to the articulated tracking of a human (one performing a "back-flip"), with the problem being the estimation of 3-D position of a defined set of proximal joints (e.g. head, torso). The dataset we use comes from a motion capture system, whereby a human is tagged with a set of infra-red reflective markers which enables one to produce a set of measurements at each of the markers. The task is to then infer, from this noisy set of measurements (at each time instance), empirical approximations to the required expectations within a skeletal graphical model structure to thus estimate the pose. This problem is a simplified version of that in reference [9], which considers the problem of pose estimation within a sequence of images. However, [9] does not allow online estimation, but does consider the more difficult problem of markerless 3D pose estimation in a video sequence. We adopt a similar potential function modelling approach as that used in [9], although the likelihood function within our model is much simpler since we do not have the additional detection (within a video sequence) to consider. We consider a constant velocity model for the motion consideration in the time axis. The estimated "skeletal" structure for several randomly selected time instances is displayed in Figure 1[4].

## 8  Conclusions and Further Work

This paper has introduced an Auxiliary Particle Belief Propagation algorithm, which is an importance sampling based approximation to the messages and the beliefs in the BP algorithm, for inference on graphical models (and so facilitating decentralised data fusion). Through the introduction of an auxiliary variable, it was shown that it is possible to reduce the computational complexity of an importance sampling approach to be linear in the number of particles, whilst also sampling from an approximation to the "optimal" importance function; optimal in the sense that the variance of the weights is minimised. This approach is prevalent in the particle filtering literature. Perhaps the most important contribution within this paper is the ability perform online inference in time evolving graphical models. This extension is very important for the design of truly distributed data fusion (and tracking system) architectures. A straightforward extension of this work would be to employ Rao-Blackwellisation, where appropriate, to further minimise

---

[4]An avi file containing the results can be downloaded from www-sigproc.eng.cam.ac.uk/∼ mb511/avifiles/SAPBPresults.avi
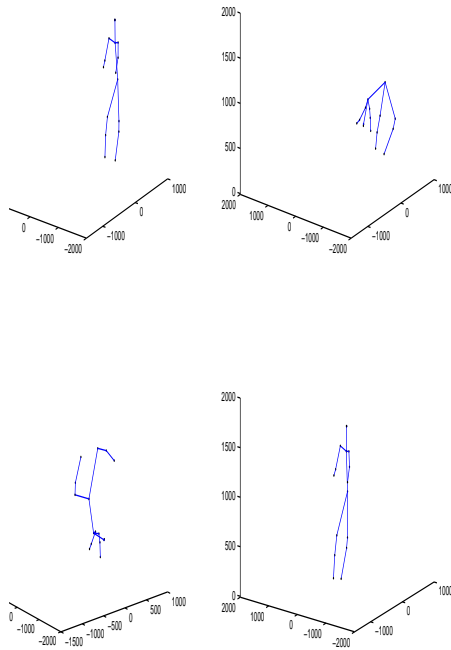
Fig. 1: Articulated object tracking results from the application of the SAPBP algorithm, (MAP estimates) displayed at 4 randomly chosen time instances.

the variance of the estimates. In addition, considering a fixed-lag over the temporal aspect would improve results.

In future work, we aim to apply this algorithm to the articulated tracking of multiple humans in a sequence of video images where data association ambiguity is present. We also aim to study decentralised tracking architectures, where one could also consider the dimension of the joint state of the targets to be a random variable itself.

## Acknowledgements

## References

[1] C Andrieu, M Davy, and A Doucet. Improved auxiliary particle filtering: Application to time-varying spectral analysis. In *IEEE Workshop on Statistical Signal Processing, Singapore*, 2001.

[2] C Berzuini, N G Best, W R Gilks, and C Larizza. Dynamic conditional independence models and Markov chain Monte Carlo methods. *Journal of the American Statistical Association*, 92:1403–1412, 1997.

[3] G Hua and Y Wu. Multi-scale visual tracking by Sequential Belief Propagation. In *Proceedings of the IEEE Computer Science Conference on Computer Vision and Pattern Recognition*, 2004.

[4] M Isard. Pampas: Real-valued graphical models for computer vision. In *Computer Vision and Pattern Recognition*, volume 1, pages 613–620, 2003.

[5] M I Jordan, editor. *Learning in graphical models*. Cambridge MA: MIT Press, 1999.

[6] K Murphy, Y Weiss, and M Jordan. Loopy-belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, July 1999.

[7] J Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.

[8] M K Pitt and N Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

[9] L Sigal, M Isard, B H Sigelman, and M Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *Advances in Neural Information Processing Systems*, 2003.

[10] E Sudderth, A Ihler, W Freeman, and A Willsky. Nonparametric Belief Propagation. Technical Report 2551, Massachusetts Institute of Technology Laboratory for Information and Decision Systems, October 2002.

[11] R van der Merwe, A Doucet, J F G de Freitas, and E Wan. The Unscented particle filter. In T K Leen, T G Dietterich, and V Tresp, editors, *Advances in Neural Information Processing Systems 13*. MIT Press, 2000.

[12] J S Yedidia, W T Freeman, and Y Weiss. Generalized belief propagation. In T K Leen, T G Dietterich, and V Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 689–695. MIT Press, 2001.