2D Articulated Tracking With Dynamic Bayesian Networks

Chunhua Shen, Anton van den Hengel, Anthony Dick, Michael J. Brooks
School of Computer Science, The University of Adelaide, SA 5005, Australia
{chhshen,anton,ard,mjb}@cs.adelaide.edu.au

Abstract

We present a novel method for tracking the motion of an articulated structure in a video sequence. The analysis of articulated motion is challenging because of the potentially large number of degrees of freedom (DOFs) of an articulated body. For particle filter based algorithms, the number of samples required with high dimensional problems can be computationally prohibitive. To alleviate this problem, we represent the articulated object as an undirected graphical model (or Markov Random Field, MRF) in which soft constraints between adjacent subparts are captured by conditional probability distributions. The graphical model is extended across time frames to implement a tracker. The tracking algorithm can be interpreted as a belief inference procedure on a dynamic Bayesian network. The discretisation of the state vectors makes it possible to utilise the efficient belief propagation (BP) and mean field (MF) algorithms to reason in this network. Experiments on real video sequences demonstrate that the proposed method is computationally efficient and performs well in tracking the human body.

1 Introduction

Articulated tracking is an important computer vision task for a variety of applications, including human machine interfaces, gesture recognition and human activity analysis for video surveillance. However the computational complexity of tracking an articulated target increases exponentially with the number of degrees of freedom (DOFs) of the target, and is further complicated by image ambiguities and self-occlusion. Exponential complexity is a severe problem when tracking highly articulated structures such as the human body, which is typically modelled with over 20 DOFs.

Many approaches have been studied to circumvent this problem. For particle filter based tracking algorithms [1], various strategies have been proposed to improve the particle filters' sampling efficiency so that fewer particles are needed to represent the filtering distribution, thereby reducing the computational cost. Such techniques include the Annealed Particle Filter [2], Hybrid Monte Carlo Filter [3], Kernel Particle Filter [4] and multiple hypothesis tracking [5]. The core ideas behind these algorithms are similar: use a stochastic (Markov chain Monte Carlo in [3]), or deterministic (hill climbing in [2, 4, 5]) optimisation method to drive the particles to the dominant modes in the likelihood or posterior distribution space. Thus fewer particles are needed to represent the distribution well. In a similar vein, the Unscented Particle Filter [6] constructs a more accurate sampling distribution from which to draw particles, thereby improving the performance of the standard particle filter.

Alternatively, several strategies have been devised specifically for articulated tracking. One is to reduce the configuration dimensionality by using specialised knowledge of the target's articulation and motion patterns (*e.g.* [7]). Another approach is to combine body part detection with tracking. If one part of the articulated model can be detected and localised at the first stage, it can be used to reduce the configuration space and update subsets of the state parameters [8]. However, such approaches are usually *ad hoc*.

Another approach is to model the articulated body by the joint probability density function of the position, velocity, or any other states of a collection of subparts (more details in Section 2). The probabilistic conditional dependence structure of subparts is encoded by an undirected graphical model such as a Markov Random Field (MRF). Based on a particle filtering (or sequential Monte Carlo) technique, the graphical model can be extended across time frames to implement a tracker [9, 10]. The tracking algorithm can be interpreted as a belief inference procedure on a dynamic Bayesian network (DBN). Two popular algorithms for probabilistic inference on such a graphical model are Belief Propagation (BP) and Mean Field (MF) methods. BP and MF are both approximations that reduce the complexity of inference. Compared with conventional particle filtering, the advantage of this approach is that it converts the exponential complexity of the conventional particle filter to linear complexity in the number of subparts. However both BP and MF operate on a discrete state space, whereas

Proceedings of the Fourth International Conference on Computer and Information Technology (CIT'04)
0-7695-2216-5/04 $20.00 © 2004 IEEE

the state vector of a tracked body is real-valued. The discretisation of the state vectors makes it possible to utilise the efficient BP and MF approximation algorithms to reason in such a graph.

The main contribution of this paper is to cast the 2D articulated tracking problem into a discrete dynamic Bayesian network framework in which the efficient BP and MF methods of inference can be adopted to implement visual tracking based on Bayesian filtering. Experiments on real video sequences demonstrate that the proposed method is computationally efficient and both of the two inference methods perform well in tracking the configuration of the human body.

2 Related Work

Articulated visual tracking has been extensively researched in recent years. In this section, we briefly cover some work which is closely related to ours. In [10] Wu *et al* describe articulated motion as a collection of the individual motions of subparts, and use an undirected graph to model the constraints between subparts. A mean field Monte Carlo (MFMC) algorithm is proposed to perform inference on the real-valued graphical model. The same basic idea for modelling an articulated body is used in [9], this time focusing on 3D human body structure recovery rather than 2D tracking. However, instead of using MFMC, a real-valued nonparametric belief propagation algorithm (NBP [11] or PAMPAS [12]) is used in [9]. In [13] the authors extend their previous work to 3D loose-limbed people tracking following the same principal. Similarly, Sudderth *et al* model hand kinematics with a graphical model and use NBP to track hand motion [14].

Because the state spaces of the graphical models in both of the above works are continuous, real-valued inference is applied. An alternative solution is to discretise the state space. This is viable for 2D tracking, due to the relatively low dimensionality of the configuration space of each individual part [15, 16]. The advantages of using a discrete graphical model are that it is more computationally efficient, and no approximation assumptions are needed.

Exact inference in densely connected graphs is computationally intractable. Hence approximate inference methods such as belief propagation [17] and variational methods (of which mean field is the simplest and the most efficient version [18]) are used to obtain a local maximum of the posterior instead of the global one.

The outline of the remaining content is as follows. In Section 3 we describe the dynamic Markov network for articulated motion. Section 4 introduces the inference algorithms, *i.e.*, the BP and MF methods. Experiments on real video sequences are presented in Section 5. Finally, concluding remarks are presented in Section 6.

3 Modelling the Articulated Body

A high level view of articulated tracking with a dynamic Bayesian network is depicted in Fig. 1. The articulated body is modelled by an undirected graphical model of N nodes with pairwise potentials, in which each graph node corresponds to a rigid subpart. We denote each individual subpart's state vector as $\mathbf{x}_i$ ($i \in [1, N]$). The entire state space is $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and the corresponding observation space is $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$. Note that in this paper, we discretise the continuous space into discrete values, hence $\mathbf{x}_i \in \mathcal{D}\,(i \in [1, N])$ where $\mathcal{D}$ is a finite, discrete set. The task is to infer the posterior $p(\mathbf{x}_{i,t+1}|\mathbf{z}_{i,1:t+1})$ at every time frame $t+1$ given $p(\mathbf{x}_{i,t}|\mathbf{z}_{i,1:t})$ and dynamical models $p(\mathbf{x}_{i,t+1}|\mathbf{x}_{i,t})$, ($i \in [1, N]$), under the Markovian assumption. Generally, for a single frame, the inference problem

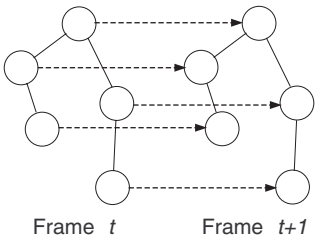

Figure 1. Dynamic Bayesian network for tracking articulated motion. Empty circles represent state nodes, each of which is associated with an observation node. observation nodes are not shown in this figure. The dash line describes the propagation of the state variables in the temporal domain.

is stated by a pairwise MRF [17],

$$p(\mathcal{X}|\mathcal{Z}) = \left(\frac{1}{Z'}\prod_{i,j\in\mathcal{S}}\psi_{ij}(\mathbf{x}_i,\mathbf{x}_j)\right)\cdot\left(\prod_i \phi_i(\mathbf{x}_i)\right), \quad (1)$$

where $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ are the potential functions between two neighbouring nodes $i, j \in \mathcal{S}$ ($\mathcal{S}$ is the neighbourhood set), and $\phi_i(\mathbf{x}_i) \equiv p(\mathbf{x}_i, \mathbf{z}_i) = p(\mathbf{x}_i)p(\mathbf{z}_i|\mathbf{x}_i)$ are the local potential functions. Z' in Eq. (1) is a normalisation constant. If the local prior $p(\mathbf{x}_i)$ is unknown or intractable, a simple hypothesis is to assume a uniform distribution density in which case the joint probability $\phi_i(\mathbf{x}_i)$ degenerates to the local likelihood.

From the definition, we see that each local potential $\phi_i(\mathbf{x}_i)$ captures the local interaction between the latent state $\mathbf{x}_i$ and the observation $\mathbf{z}_i$, while the correlation potential $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ captures the soft constraints between two connected parts. The correlation potentials $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ can be modelled with a simple Gaussian distribution [10], a Gaussian with outliers [12] or a Gaussian mixture with outliers

[9], depending on the complexity of the application. In this paper, we follow [10] and use a Gaussian to model the potentials:

$$\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \propto \mathcal{N}(\mathbf{x}_j; \mu_{ij}, \Sigma_{ij}) \tag{2}$$

where μ_{ij} and Σ_{ij} are the mean and covariance of the Gaussian process respectively. They can be determined by the training data.

Given the local and correlation potentials, message passing algorithms (both the BP and MF method can be classified in this category) can be used to iteratively compute the belief at each state node. This is described in the following section.

4 Inference Algorithms

4.1 Inference with Belief Propagation and Mean Field Approximation

The concept of a *message* explains intuitively the inference procedure of BP. The message $m_{ji}(\mathbf{x}_i)$ is sent from the node j to i ($j \rightarrow i$). The belief propagation algorithm iterates [19]:

$$m_{ji}(\mathbf{x}_i) \leftarrow \alpha \sum_{\mathbf{x}_j} \left(\psi_{ji}(\mathbf{x}_j, \mathbf{x}_i) \phi_j(\mathbf{x}_j) \prod_{k \in \mathcal{S}(j) \backslash i} m_{kj}(\mathbf{x}_j) \right) \tag{3}$$

where $\mathcal{S}(j)\backslash i$ represents all the neighbouring nodes of $\mathbf{x}_j$ except $\mathbf{x}_i$. The belief (the marginal probability) at the node $\mathbf{x}_i$ is

$$b_i(\mathbf{x}_i) \leftarrow \alpha \phi_i(\mathbf{x}_i) \prod_{j \in \mathcal{S}(i)} m_{ji}(\mathbf{x}_i). \tag{4}$$

Note that α is a normalisation constant so that $b_i(\mathbf{x}_i)$ satisfies the normalisation constraint $\sum_{\mathbf{x}_i} b_i(\mathbf{x}_i) = 1$.

The MF approximation is obtained by minimising the Kullback-Leibler divergence between a fully factorised distribution $q(\mathcal{X}) = \prod_i b_i(\mathbf{x}_i)$ and the distribution $p(\mathcal{X}|\mathcal{Z})$ (for more details refer to [20]). It involves a similar update strategy:

$$b_i(\mathbf{x}_i) \leftarrow \alpha \phi_i(\mathbf{x}_i) \exp \left(\sum_{j \in \mathcal{S}(i)} \sum_{\mathbf{x}_j} b_j(\mathbf{x}_j) \log \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \right) \tag{5}$$

where α normalises the beliefs.

The pattern of message passing in these two update procedures is different. In BP, every node $\mathbf{x}_i$ sends a different message m_{ij} to its neighbours. This message itself is a collection of messages received from all the *other* neighbours in the previous iteration. In MF, every node $\mathbf{x}_i$ sends a single message b_i to its neighbours based on the messages it received from *all* of its neighbours in the previous iteration. The relationship between these two methods has been explored in [18].

4.2 Tracking with Neighbourhood Constraints

Section 4.1 describes the BP and MF algorithms at one time frame. In this section, we use these two algorithms to incorporate constraints between neighbouring nodes (or *neighbourhood constraints*) into a tracking algorithm. The tracking problem is formulated as Bayesian filtering [1]:

$$p(\mathcal{X}_t|\mathcal{Z}_{1:t}) \propto p(\mathcal{Z}_t|\mathcal{X}_t) \int p(\mathcal{X}_t|\mathcal{X}_{t-1}) p(\mathcal{X}_{t-1}|\mathcal{Z}_{1:t-1}) d\mathcal{X}_{t-1}.$$

There are three components involved in this filtering procedure: the previous posterior $p(\mathcal{X}_{t-1}|\mathcal{Z}_{1:t-1})$, the system dynamics $p(\mathcal{X}_t|\mathcal{X}_{t-1})$, and the likelihood $p(\mathcal{Z}_t|\mathcal{X}_t)$. We add another component, the neighbourhood constraints, into the filtering.

Due to the nonlinear non-Gaussian nature of most real tracking applications, we track using a particle filter. However, if the dimensionality of the state $\mathcal{X}$ is large, a prohibitive number of particles are needed to effectively sample the space of $\mathcal{X}$. Our model partitions the state into rigid parts and reduces the complexity from being exponential with respect to the dimensionality of $\mathcal{X}$ to being linear in the number of parts. However rather than just using multiple independent Kalman/particle filters to track each part, our model takes constraints between neighbouring parts into consideration.

We assume that the likelihood of each subpart is independent, hence

$$p(\mathcal{Z}|\mathcal{X}) = \prod_{i=1}^{N} p_i(\mathbf{z}_i|\mathbf{x}_i). \tag{6}$$

At time frame t, the belief for the i-th part is written [1] (in BP and MF, the belief $b_{i,t}(\mathbf{x}_{i,t})$ is an approximation of the posterior $p(\mathbf{x}_{i,t}|\mathcal{Z}_{1:t})$),

$$b_{i,t}(\mathbf{x}_{i,t}) \propto \phi_{i,t}(\mathbf{x}_{i,t}) \int p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}) b_{i,t-1}(\mathbf{x}_{i,t-1}) d\mathbf{x}_{i,t-1} \times M_{i,t}(\mathbf{x}_{i,t}), \tag{7}$$

where $M_{i,t}(\mathbf{x}_{i,t})$ is defined as the neighbourhood constraints at node i at time frame t. As its name implies, the neighbourhood constraints denote the soft constraints contributed by a node's neighbours. Comparing Eq. (7) to Eq. (3),(4) and Eq. (5), we write the neighbourhood constraints as [2] :

BP algorithm

$$M_i(\mathbf{x}_i) = \prod_{j \in \mathcal{S}(i)} m_{ji}(\mathbf{x}_i) \tag{8}$$

[1] In [10], Wu *et al* use a similar equation, the only difference is that they describe the neighbourhood constraints in the continuous space.

[2] For clarity, the subscript t, indicating the time frame, is omitted in Eq. (8) and Eq. (9).

where m_{ji} is calculated by Eq. (3);

MF algorithm

$$M_i(\mathbf{x}_i) = \exp\left(\sum_{j\in\mathcal{S}(i)}\sum_{\mathbf{x}_j} b_j(\mathbf{x}_j)\log\psi_{ij}(\mathbf{x}_i,\mathbf{x}_j)\right). \quad (9)$$

The new tracking schema is shown in Fig. 2, a graphical representation of Eq. (7). Compared with standard Bayesian filtering, some additional neighbouring constraints are introduced into the model. According to Fig. 2, it is quite straightforward to deduce Eq. (7). The posterior density is propagated along the temporal axis by convolving with the dynamical model. Therefore, at time t, node i receives the previous posterior $b_{i,t-1}(\mathbf{x}_{i,t-1})$ which is then updated by the local likelihood $\phi_{i,t}(\mathbf{x}_{i,t})$ as well as the neighbourhood constraints $M_{i,t}(\mathbf{x}_{i,t})$. The whole framework can

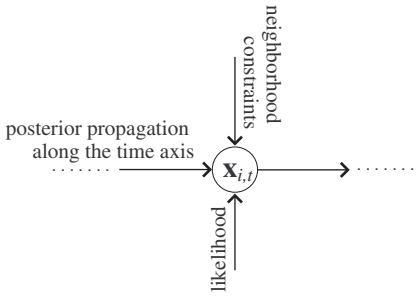

Figure 2. Neighbourhood constraints are combined into the conventional Bayesian filter.

be regarded as a constrained system of Bayesian filters, in which multiple Bayesian filters are linked by some soft constraints. From another viewpoint, as pointed out in the previous sections, this model is also an extension of the undirected graphical model—it extends the undirected graphical model along the temporal axis.

A contradiction arises concerning the state space. In previous sections, the calculation of the neighbourhood constraints was based on the discrete BP or MF methods, while the tracking is usually processed in a continuous space. One solution is to calculate the neighbourhood constraints in a continuous space using NBP [11], PAMPAS [12] or MFMC [10]. Both NBP and PAMPAS model distributions as Gaussian mixtures. An MCMC sampler is then used to sample from the product of Gaussian mixture distributions, which is computationally intensive. MFMC avoids sampling directly from the product of Gaussian mixtures because only single-dimensional integrals are involved. It is completely nonparametric and generates samples by importance sampling. However it is not a trivial task to design importance sampling functions without confident domain knowledge. Alternatively tracking can be formulated in a discrete state space so that dynamic programming (DP) techniques such as a hidden Markov model (HMM) [21] can be used. Although DP is efficient, the quantisation error introduced by the discretisation is potentially large.

As a novelty, we combine discrete and continuous inference in the same framework. In order to minimise the quantisation error, the posterior is propagated along the temporal axis in a continuous state space via particle filtering[3] while the neighbourhood constraints which propagate among neighbouring nodes are calculated in a discrete space with BP or MF. Such an approach is a compromise between quantisation error and computational efficiency. It is based on the following two facts. Firstly, from Fig. 2 we know that there are four components which affect the tracking, namely, the previous posterior, the dynamic model, the local likelihood and the neighbourhood constraints. The calculation of just the neighbourhood constraints in a discrete space introduces less quantisation error compared with discretising the whole algorithm. Secondly, the computation of the neighbourhood constraints in the continuous space is much more complex than discrete inference. An overview of the entire tracking algorithm, using the MF and BP methods, is given in Fig. 3. The procedures for updating $M_i(\mathbf{x}_i)$ in MF and BP are slightly different, although the essentials are similar. The difference is that for BP, the update rules for the neighbourhood constraints do not depend explicitly on the estimation of the belief (posterior), whereas the MF updates for the neighbourhood constraints depend on the belief, which means the estimates of the belief must be re-estimated iteratively.

In step 2 of the importance sampling procedure, it can save some computation to replace $\mathbf{x}_{i,t}^{(n)}$ with $\widetilde{\mathbf{x}}_{i,t}^{(n)}$ because the message passing step is processed in the discrete space. Here 'closest' means the minimum weighted Euclidean distance. We give the third parameter (angle) more weight because, empirically, the orientation of a part is more important than its position. Other distance metrics may also be applicable.

We use a different methodology to [10] to introduce neighbourhood constraints into particle filtering. The neighbourhood constraints in the sequential message passing algorithm are formulated as an additional factor in the importance sampling step. By contrast, in [10] the MFMC algorithm has two steps. The first step is particle sampling, after which the particles are transferred to the message passing process which iterates to convergence.

5 Evaluation

In this section, the effectiveness of the proposed framework is tested on several real videos. In our experiments,

[3] Kalman filters can also be adopted. We will use Unscented Kalman filters to propose a different approach. In this paper, we use particle filters.

- *Start tracking*: Set $t = 1$. For i-th subpart ($i = 1, \ldots, N$), sample from the prior to generate N_0 samples $\{\mathbf{x}_{i,t-1}^{(n)}, \omega_{i,t-1}^{(n)}\}_{n=1}^{N_0}$.
- *Re-sampling*: Re-sample to obtain N_0 replacement particles $\{\mathbf{x}_{i,t-1}^{(n)}, \frac{1}{N_0}\}_{n=1}^{N_0}$ according to the weights $\omega_{i,t-1}^{(n)}$.
- *Importance sampling*:

  1. For $n = 1, \ldots, N_0$, sample $\{\mathbf{x}_{i,t}^{(n)}, \omega_{i,t}^{(n)}\}_{n=1}^{N_0} \sim f(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}^{(n)}, \mathbf{z}_{i,1:t})$ where $f(\cdot)$ is the importance function.
  2. In the discretised space, find $\widetilde{\mathbf{x}}_{i,t}^{(n)}$ which is closest to $\mathbf{x}_{i,t}^{(n)}$. Set $\mathbf{x}_{i,t}^{(n)} = \widetilde{\mathbf{x}}_{i,t}^{(n)}$.
  3. Message Passing Process:

  I. Belief Propagation:

  (a) Set $\kappa = 0$. Initialise all messages $m_{ji,t}^{\kappa}(\mathbf{x}_{j,t})$, $(i, j \in [1, N])$ with a uniform distribution.

  (b) Update all messages. Iterate until convergence:

  $$\kappa = \kappa + 1,$$
  $$m_{ji,t}^{\kappa}(\mathbf{x}_{i,t}^{(n)}) \leftarrow \alpha \sum_{\mathbf{x}_{j,t}} \left(\psi_{ij}(\mathbf{x}_{i,t}^{(n)}, \mathbf{x}_{j,t}) \phi_{j,t}(\mathbf{x}_{j,t}) \prod_{k \in \mathcal{S}(j) \setminus i} m_{kj,t}^{\kappa-1}(\mathbf{x}_{j,t})\right).$$

  (c) Calculate the neighbourhood constraints:

  $$M_{i,t}(\mathbf{x}_{i,t}^{(n)}) = \prod_{j \in \mathcal{S}(i)} m_{ji,t}(\mathbf{x}_{i,t}^{(n)}).$$

  II. Mean Field:

  (a) Set $\kappa = 0$. Initialise all messages $b_{i,t}^{\kappa}(\mathbf{x}_{i,t}) = \phi_{i,t}(\mathbf{x}_{i,t})$, $i \in [1, N]$.

  (b) Update all messages. Iterate until convergence:

  $$\kappa = \kappa + 1,$$
  $$b_{i,t}^{\kappa}(\mathbf{x}_{i,t}^{(n)}) \leftarrow \alpha \phi_{i,t}(\mathbf{x}_{i,t}^{(n)}) \exp\left(\sum_{j \in \mathcal{S}(i)} \sum_{\mathbf{x}_{j,t}} b_{j,t}^{\kappa-1}(\mathbf{x}_{j,t}) \log \psi_{ij}(\mathbf{x}_{i,t}^{(n)}, \mathbf{x}_{j,t})\right).$$

  (c) Calculate the neighbourhood constraints:

  $$M_{i,t}(\mathbf{x}_{i,t}^{(n)}) = \exp\left(\sum_{j \in \mathcal{S}(i)} \sum_{\mathbf{x}_{j,t}} b_{j,t}^{\kappa}(\mathbf{x}_{j,t}) \log \psi_{ij}(\mathbf{x}_{i,t}^{(n)}, \mathbf{x}_{j,t})\right).$$

  4. Re-weight:

  $$\omega_{i,t}^{(n)} = \frac{\phi_{i,t}(\mathbf{x}_{i,t}^{(n)}) p(\mathbf{x}_{i,t}^{(n)}|\mathbf{x}_{i,t-1}^{(n)}) M_{i,t}(\mathbf{x}_{i,t}^{(n)})}{f(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}^{(n)}, \mathbf{z}_{i,1:t})}.$$

  Normalise the weights. We obtain the posterior distribution represented by weighted particles $\{\mathbf{x}_{i,t}^{(n)}, \omega_{i,t}^{(n)}\}_{n=1}^{N_0}$.
- Set $t = t + 1$, go to the *Re-sampling* step to process the next frame.

Figure 3. The Sequential Belief Propagation algorithm (SBP) and the Sequential Mean Field algorithm (SMF)—in practice either algorithm can be used for articulated tracking. The symbols n, i, t, κ indicate the particle label, subpart label, time frame and message passing iteration, respectively.

we focus on 2D tracking. We use a "cardboard" model in which the body is represented by planar patches [22]. Each patch is an isosceles trapezoid which is determined by the length of its sides l_1, l_2, l_3, the position of its centre (x_0, y_0) and its orientation θ. In our experiments, we fix the length of each side. Thus each part has three degrees of freedom. For simplicity, we use a first order auto-regression (AR) equation to model the system dynamics. This has the form $\mathbf{x}_{i,t} = \mathbf{C}\mathbf{x}_{i,t-1} + \mathbf{D}\mathbf{u}_t$ where $\mathbf{u}_t$ is a multivariate normal distribution, the matrix $\mathbf{C}$ defines the deterministic component and $\mathbf{D}$ the stochastic component.

Another important factor in tracking is the likelihood $p(\mathbf{z}_{i,t}|\mathbf{x}_{i,t})$. In this paper, we use only a colour feature due to its modest computational cost, although edge detection is a promising and frequently used cue in visual contour tracking [1]. Motion information is another useful cue in tracking; we will improve the performance of this tracker by including multiple cues as in [23]. We follow the non-parametric model presented in [24] to implement the colour tracker. Colour histograms are calculated in the RGB space by a modified mean shift algorithm as in [24]. In our experiments, we find that $8 \times 8 \times 8$ bins are sufficient to represent the colour distribution for pixels with 8-bit colour depth in

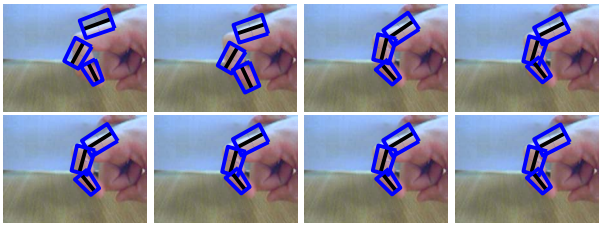

Figure 4. The first 6 iteration results obtained by the SMF algorithm on the Finger image sequence. The first image is the initial state of the each part. The second image is the state obtained by the independent particle filters. (*i.e.* merely considering the likelihood correction when using the dynamic prior as the proposal distribution for importance sampling.) The final 6 images are the results of the first 6 MF iterations, demonstrating rapid convergence. We obtain quite similar results with SBP algorithm because the graph structure is a simple chain without loops.

each channel. The target colour model for each subpart is initialised by hand and it is not updated during the tracking. The resolution of all the images is 320×240.

Some intermediate iteration results on a 3-part finger image sequence are shown in Fig. 4. Before iteration, the initial state (*i.e.* the mean state of the propagated previous posterior by applying the dynamic model) of the tracked finger is far from the correct position. After the likelihood correction process, the state (see the second image Fig. 4) is still not satisfactory and the orientation angle of each part is not corrected. This is because the fingers' skin colour distribution is almost uniform and no edge information is utilised in this experiment. As expected, the proposed algorithm leads to the right positions by applying the soft constraints between each linked articulated part.

In this 3-part tracking experiment, the proposed algorithm converges very quickly. From the second iteration on, only slight amelioration is observed. The distance between each joint point becomes smaller and more natural after more iterations. The reason is that the graph structure corresponding to the 3-part finger is quite simple.

We have obtained similar tracking results with both SBP and SMF algorithm in our experiments, again due to the simple graph structure. The difference between the BP and MF is merely the degree of the approximation used [25]. In BP, the target distribution is represented by a Bethe approximation while in MF, it is approximated by a lower order factorisable distribution. For simple graphs such as chains or trees, these two approximations will be comparable (for a chain, BP is exact which yields the correct marginals, whereas MF might provide incorrect marginals). It is expected that SBP will surpass SMF when tracking those articulated objects whose corresponding structures are complex (*e.g.* loopy graphs). In the human body tracking, the limbs' self-occlusion could be modelled by adding additional connections between those occluded limbs in the Bayesian network. In such cases, we have to infer in loopy graphs in which SBP is supposed to outperform SMF.

The second test video is a 10-part body motion sequence whose corresponding graph is a tree. We compared the results with SMF (the results with SBP are similar) and multiple independent particle filters. Fig. 5 and Fig. 6 depict results. Due to the complex motion of the full body, the multiple independent particle filters lose the tracked subparts easily after several frames, while the proposed algorithm can track them successfully.

In the algorithm implementation, for the sake of the efficiency, we abandon those states whose belief is lower than a predefined threshold. Thus far fewer states need to be taken into consideration in each iteration. In fact, only those states around the state estimated by the conventional particle filter are considered. Experiments show such an approach is feasible. We will discuss how to speed up the algorithm in the last section.

6 Discussion and Conclusion

We propose a Bayesian network model to track an articulated body in which both spatial and temporal constraints are taken into consideration. Belief propagation and mean field method are adopted for inference in the Bayesian network. Promising experimental results on real videos prove the effectiveness of this framework. Some avenues for future work are listed below. This work focuses on effective

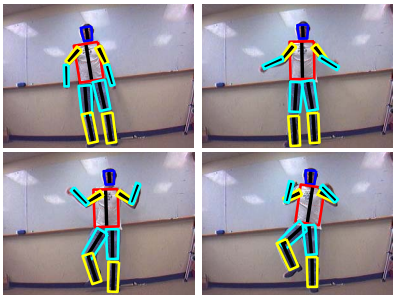

Figure 5. Human body tracking results with SMF.

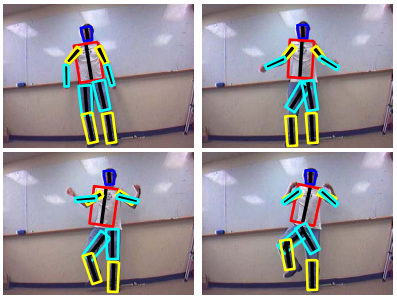

Figure 6. Human body tracking results with multiple independent particle filters.

2D tracking which is appropriate for applications such as video surveillance. We will extend it to 3D articulated structure recovery and tracking. When modelling 3D structures, the dimensionality increases and it can be difficult to discretise. In this case, real-valued belief propagation methods [10, 11, 12] should be used.

For real-time application, it is crucial to decrease the computational burden. In [16], to speed up the BP algorithm, a pruning procedure and a novel "focused message updating" strategy is proposed. By combining these two strategies, the updating speed increases considerably. These techniques can also be applied in our tracking framework to make the algorithm more efficient. An alternative to BP,

the Concave-Convex Procedure (CCCP) has been shown to outperform BP in convergence speed and stability [25]. We will explore these new algorithms in the context of articulated tracking in the future.

Another possible way to improve the performance is to integrate domain knowledge into this general tracking framework. For example, to track a walking human body, due to its relatively simple cycle motion patterns, we can learn each articulated part's motion from training data [26] rather than preset the parameters of the motion model as in this paper. A body part detector can also be augmented to initialise the tracking and recover from temporary tracking failures as in [15].

References

[1] M. Isard and A. Blake, "CONDENSATION – Conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[2] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[3] K. Choo and D. Fleet, "People tracking using hybrid Monte Carlo filtering," in *IEEE International Conference on Computer Vision*, Vancouver, Canada, 2001, vol. 2, pp. 321–328.

[4] C. Chang and R. Ansari, "Kernel particle filter: Iterative sampling for efficient visual tracking," in *IEEE International Conference on Image Processing*, Barcelona, Spain, 2003.

[5] T.-J. Cham and J. M. Rehg, "A multiple hypothesis approach to figure tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, 1999, vol. 2, pp. 239–245.

[6] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, December 2001, vol. 2, pp. 786–793.

[7] K. Rohr, "Human movement analysis based on explicit motion models," in *Motion-Based Recognition*, M. Shah and R. Jain, Eds., chapter 8, pp. 171–198. Kluwer Academic Publishers, 1997.

[8] M.-W. Lee, I. Cohen, and S.-K. Jung, "Particle filter with analytical inference for human body tracking," in *Workshop on Motion and Video Computing (MOTION'02)*, 2002.

[9] L. Sigal, M. I. Isard, B. H. Sigelman, and M. J. Black, "Attractive people: Assembling loose-limbed models using non-parametric belief propagation," in *Advances in Neural Information Processing Systems 16*, Vancouver, Canada, December 2003.

[10] Y. Wu, G. Hua, and T. Yu, "Tracking articulated body by dynamic Markov network," in *IEEE International Conference on Computer Vision*, Nice, France, 2003, pp. 1094–1101.

[11] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky, "Nonparametric belief propagation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, vol. 1, pp. 605–612.

[12] M. Isard, "PAMPAS: Real-valued graphical models for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, vol. 1, pp. 613–620.

[13] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," in *IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004.

[14] E. Sudderth, M. Mandel, W. Freeman, and A. Willsky, "Visual hand tracking using nonparametric belief propagation," in *Workshop on Generative Model Based Vision, in conjunction with CVPR 2004*, Washington, DC, 2004.

[15] D. Ramanan and D. A. Forsyth, "Finding and tracking people from the bottom up," in *IEEE Conference on Computer Vision and Pattern Recognition*, Wisconsin, 2003.

[16] J. M. Coughlan and S. J. Ferreira, "Finding deformable shapes using loopy belief propagation," in *7th European Conference on Computer Vision*, 2002, pp. 453–468.

[17] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.

[18] Y. Weiss, "Comparing the mean field method and belief propagation for approximate inference in MRFs," in *Advanced Mean Field Methods : Theory and Practice*, M. Opper and D. Saad, Eds. The MIT Press, 2001.

[19] J. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Advances in Neural Information Processing Systems*, 2000, pp. 689–695.

[20] T. Jaakkola, *Variational Methods for Inference and Estimation in Graphical Models*, Ph.D. thesis, MIT, 1997.

[21] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1518–1569, June 2002.

[22] S. X. Ju, M. J. Black, and Y. Yacoob, "Cardboard people: A parameterized model of articulated image motion," in *International Conference on Automatic Face and Gesture Recognition*, Killington, Vermont, 1996, pp. 38–44.

[23] C. Shen, A. van den Hengel, and A. Dick, "Probabilistic multiple cue integration for particle filter based tracking," in *International Conference on Digital Image Computing: Techniques and Applications*, Sydney, 2003, vol. 1, pp. 399–408.

[24] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *7th European Conference on Computer Vision*, Copenhagen, Denmark, 2002, vol. 2350 of *Lecture Notes in Computer Science*, pp. 661–675, Springer.

[25] A.L. Yuille, "CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation," *Neural Computation*, vol. 14, no. 7, pp. 1691–1722, 2002.

[26] A. Elgammal, V. Shet, Y. Yacoob, and L. Davis, "Learning dynamics for exemplar based gesture recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, Wisconsin, 2003.