

Articulated Body Motion Capture by Annealed Particle Filtering

Jonathan Deutscher
University of Oxford
Dept. of Engineering Science
Oxford, OX13PJ
United Kingdom
jdeutsch@robots.ox.ac.uk

Andrew Blake
Microsoft Research
1 Guildhall St,
Cambridge, CB2 3NH
United Kingdom
ablake@microsoft.com

Ian Reid
University of Oxford
Dept. of Engineering Science
Oxford, OX13PJ
United Kingdom
ian@robots.ox.ac.uk

Abstract

The main challenge in articulated body motion tracking is the large number of degrees of freedom (around 30) to be recovered. Search algorithms, either deterministic or stochastic, that search such a space without constraint, fall foul of exponential computational complexity. One approach is to introduce constraints — either labelling using markers or colour coding, prior assumptions about motion trajectories or view restrictions. Another is to relax constraints arising from articulation, and track limbs as if their motions were independent. In contrast, here we aim for general tracking without special preparation of subjects or restrictive assumptions.

The principal contribution of this paper is the development of a modified particle filter for search in high dimensional configuration spaces. It uses a continuation principle, based on annealing, to introduce the influence of narrow peaks in the fitness function, gradually. The new algorithm, termed annealed particle filtering, is shown to be capable of recovering full articulated body motion efficiently.

1. Introduction

Marker-based human motion capture has been used commercially [19] for a number of years with applications found in special effects and biometrics. The use of markers however is intrusive, necessitates the use of expensive specialised hardware and can only be used on footage taken especially for that purpose. A markerless system of human motion capture could be run using conventional cameras and without the use of special apparel or other equipment. Combined with today's powerful off-the-shelf PC's, cost-effective and real-time markerless human motion capture has for the first time become a possibility. Such a system would have a greater number of applications than its marker based predecessor ranging from intelligent surveillance to character animation and computer interfacing. For this reason the field of human motion capture has recently seen somewhat of a renaissance.

Research into human motion capture has so far failed to

produce a full-body tracker general enough to handle realistic real-world applications. This gives an insight into the difficulty of the problem. Research has concentrated on the articulated-model based approach. The reason this approach is popular is the high level output it produces in the form of a model configuration for each frame. This output can easily be used by higher-order processes to perform tasks such as character animation.

The problem with using articulated models is the high dimensionality of the configuration space and the exponentially increasing computational cost that results. A realistic articulated model (see figure 4) of the human body usually has at least 25 DOF. The model used in this paper for example has 29 DOF, and models employed for commercial character animation usually have over 40.

A number of effective 2D systems have been presented [7] [10]. These are good for applications such as surveillance, however they do not provide output in the form of 3D model configurations that are needed for applications such as 3D character animation.

There are several possible strategies for reducing the dimensionality of the configuration space. Firstly it is possible to restrict the range of movement of the subject. This approach has been pursued by Hogg [8], Rohr [17] and Niyogi [15]. All three assume the subject is walking. Rohr even reduces the dimension of the problem to the phase of the walking cycle. Goncalves [6] and Deutscher [3] assume a constant angle of view of the subject as does Bregler [2] and Rehg [16]. Such an approach greatly restricts the resulting trackers generality.

Another way to constrain the configuration space is to perform a hierarchical search. If one part of an articulated model can be localised independently then it can be used as a constraint for reducing the rest of the model. Gavrilu [4] does just this when he uses what he terms *search space decomposition*. He is able to localise the torso using colour cues and uses this information to constrain the search for the limbs. Without the assistance of colour cues (or other labelling cues) however it is very hard to independently localise specific body parts in realistic scenarios. This is mainly due to the problem of self occlusion and rules out

the use of a hierarchical search.

For a practical full body tracker to be developed it cannot rely on assumptions about motion, angle of view or the availability of labelling cues. The principal contribution of this paper is the development of a modified particle filter for searching high dimensional configuration spaces which does not rely on such assumptions. It uses a continuation principle, based on annealing, to introduce the influence of narrow peaks in the fitness function, gradually. The new algorithm, termed *annealed particle filtering*, is shown to be capable of recovering full articulated body motion efficiently.

2. Particle filters

Particle filtering (also known as the Condensation algorithm [9]) provides a robust Bayesian framework for human motion capture. The Condensation algorithm was developed for tracking objects in clutter, in which the posterior density $p(\mathbf{X}|\mathbf{Z}_k)$ and the observation process $p(\mathbf{Z}_k|\mathbf{X})$ are often non-Gaussian or even multi-modal (\mathbf{X} denotes the model's configuration vector, $\mathbf{Z}_k = \{\mathbf{Z}_1, \dots, \mathbf{Z}_k\}$ notates the history of observations at time t_k). The complicated nature of the observation process during human motion capture causes the posterior density to be non-Gaussian and multi-modal as shown by Deutscher [3]. It is well known that a Kalman filter will fail in this case. Deutscher *et al* were able to show that the use of a particle filter will improve tracking performance.

The posterior density $p(\mathbf{X}|\mathbf{Z}_k)$ is represented by a set of weighted particles $\{(s_k^{(0)}, \pi_k^{(0)}) \dots (s_k^{(N)}, \pi_k^{(N)})\}$ where the weights $\pi_k^{(n)} \propto p(\mathbf{Z}_k|\mathbf{X} = s_k^{(n)})$ are normalised so that $\sum_N \pi_k^{(n)} = 1$. The state \mathcal{X}_k at each time step t_k can be estimated by

$$\mathcal{X}_k = \mathcal{E}_k[\mathbf{X}] = \sum_{n=1}^N \pi_k^{(n)} s_k^{(n)} \quad (1)$$

or the mode

$$\mathcal{X}_k = \mathcal{M}_k[\mathbf{X}] = s_k^{(j)}, \pi_k^{(j)} = \max(\pi_k^{(n)}) \quad (2)$$

of the posterior density $p(\mathbf{X}|\mathbf{Z}_k)$.

Particle filtering works well because it can model uncertainty. Less likely model configurations will not be thrown away immediately but given a chance to prove themselves later on, resulting in more robust tracking. However a price is paid for these attributes in computational cost. The most expensive operation in the standard Condensation algorithm is an evaluation of the likelihood function $p(\mathbf{Z}_k|\mathbf{X} = s_k^{(n)})$ and this has to be done once at every time step for every particle. To maintain a fair representation of $p(\mathbf{X}|\mathbf{Z}_k)$ a certain number of particles are required, and this number grows with the size of the model's configuration space. In fact it has been shown by MacCormick and Blake [14] that

$$N \geq \frac{\mathcal{D}_{min}}{\alpha^d} \quad (3)$$

where N is the number of particles required, d is the number of dimensions. The survival diagnostic \mathcal{D}_{min} and the particle survival rate α are both constants with $\alpha \ll 1$. An explanation of both of these constants can be found in section 5. Clearly when d is large normal particle filtering becomes infeasible.

Partitioned sampling was developed by MacCormick and Blake [13] as a variation on Condensation to reduce the number of particles needed to track more than one object. MacCormick [14] has also now applied this technique to tracking articulated objects. Using partitioned sampling reduces the number of particles required to

$$N >= \frac{\mathcal{D}_{min}}{\alpha}. \quad (4)$$

making the problem tractable. However, this assumes that the configuration space can be sliced so that one can construct an observation density $p(\mathbf{Z}_k|x_k^i)$ for each dimension x_k^i of the model configuration vector $\mathbf{X} = \{x_k^0 \dots x_k^d\}$. This assumption, that it is possible to independently localise separate parts of an articulated model, is similar to that made by Gavrilu to enable a hierarchical search. It has already been argued that it is not possible to use this approach without the use of labelling cues.

Another variation on the standard particle filter used to reduce the number of particles needed to effectively represent a posterior density has been developed by Sullivan *et al* [18]. Called *layered sampling* it is centered around the concept of importance resampling. Experimental evidence however suggests that this technique is not sufficient to solve the problem of tracking with $d > 30$, reducing the number of particles required by at best a factor of 5 to 10 before the expected behaviour of the Condensation framework breaks down.

The second reason why Bayesian particle filtering may not be suitable for full body human motion capture is the difficulties associated with constructing a valid observation model $p(\mathbf{Z}_k|\mathbf{X}_k)$ as a normalised probability density distribution. Another factor is the computational cost of calculating $p(\mathbf{Z}_k|\mathbf{X} = s_k^{(n)})$. Often an intuitive weighting function $w(\mathbf{Z}_k, \mathbf{X})$ can be constructed that approximates the probabilistic likelihood $p(\mathbf{Z}_k|\mathbf{X}_k)$ but which requires much less computational effort to evaluate. Probabilistic observation models also have a tendency to utilise only the information that can be modelled well, discarding other available information.

Given these factors it was decided to reduce the problem from propagating the conditional density $p(\mathbf{X}|\mathbf{Z}_k)$ using $p(\mathbf{Z}|\mathbf{X})$ to finding the configuration \mathcal{X}_k which returns the maximum value from a simple and efficient weighting function $w(\mathbf{Z}_k, \mathbf{X})$ at each time t_k , given \mathcal{X}_{k-1} . By doing this gains will be made on two fronts. It should be possible to make do with fewer likelihood (or weighting function) evaluations because the function $p(\mathbf{X}|\mathbf{Z}_k)$ no longer has to be fully represented and an evaluation of a simple weighting

function $w(\mathbf{Z}_k, \mathbf{X})$ should require minimal computational effort when compared to an evaluation of the observation model $p(\mathbf{Z}_k|\mathbf{X})$. The main disadvantage will be not being able to work within a robust Bayesian framework.

It was decided to continue to use a particle based stochastic framework because of its ability to handle multi-modal likelihoods, or in the case of a weighting function, one with many local maxima. The question is: *What is an efficient way to perform a particle based stochastic search for the global maximum of a weighting function with many local maxima?* It was decided to use an approach which is similar to that of simulated annealing.

3. Simulated annealing

The Markov chain based method of simulated annealing was developed by Kirkpatrick *et al* [11] as a way of handling multiple modes in an optimisation context. It employs a series of distributions, with probability densities given by $p_0(x)$ to $p_M(x)$, in which each $p_m(x)$ differs only slightly from $p_{m+1}(x)$. Samples actually need to be drawn from the distribution $p_0(x)$. The distribution p_M is designed so that the Markov chain used to sample from it allows movement between all regions of the state/search space. The usual method is to set $p_m(x) \propto p_0(x)^{\beta_m}$, for $1 = \beta_0 > \beta_1 > \dots > \beta_M$.

An annealing run is started in some initial state, from which a Markov chain designed to converge to p_M is first simulated. Some number of iterations of a Markov chain designed to converge to p_{M-1} are simulated next, starting from the final state of the previous simulation. This process is continued in this fashion, using the final state of the simulation for p_m as the initial state for the simulation for p_{m-1} , until the chain designed to converge to p_0 is finally simulated.

Note that if p_0 contains isolated modes, simply simulating the Markov chain designed to converge to p_0 starting from some arbitrary point could give very poor results, as it might become stuck in whatever mode is closest to the starting point, even if that mode has little of the total probability mass. The annealing process is a heuristic for avoiding this, by taking advantage of the freer movement possible under the other distributions. This is exactly the kind of behaviour needed for the stochastic search. One wants to move towards the global maximum of the weighting function $w(\mathbf{Z}_k, \mathbf{X})$, using the overall trend of the matching function as a guide, without becoming misguided by local maxima as seen in figure 1.

The idea of annealing for optimisation is now adapted to perform a particle based stochastic search within the framework of an annealed particle filter.

4. Annealed particle filter

A series of weighting functions $w_0(\mathbf{Z}, \mathbf{X})$ to $w_M(\mathbf{Z}, \mathbf{X})$ are employed in which each w_m differs only slightly from w_{m-1} (see figure 2, where $M = 3$). The function w_M is designed

to be very broad, representing the overall trend of the search space while w_0 should be very peaked, emphasising local features. This is achieved by setting

$$w_m(\mathbf{Z}, \mathbf{X}) = w(\mathbf{Z}, \mathbf{X})^{\beta_m}, \quad (5)$$

for $\beta_0 > \beta_1 > \dots > \beta_M$, where $w(\mathbf{Z}, \mathbf{X})$ is the original weighting function. Because it is not the aim to sample from $w(\mathbf{Z}, \mathbf{X})$, but only to find its maximum it is not required that $\beta_0 = 1$.

One annealing run is performed at each time t_k using image observations \mathbf{Z}_k . The state of the tracker after each layer m of an annealing run is represented by a set of N weighted particles

$$\mathcal{S}_{k,m}^\pi = \{(\mathbf{s}_{k,m}^{(0)}, \pi_{k,m}^{(0)}) \dots (\mathbf{s}_{k,m}^{(N)}, \pi_{k,m}^{(N)})\}. \quad (6)$$

An unweighted set of particles will be denoted

$$\mathcal{S}_{k,m} = \{(\mathbf{s}_{k,m}^{(0)}) \dots (\mathbf{s}_{k,m}^{(N)})\}. \quad (7)$$

Each particle in the set $\mathcal{S}_{k,m}^\pi$ is considered as an $(\mathbf{s}_{k,m}^{(i)}, \pi_{k,m}^{(i)})$ pair in which $\mathbf{s}_{k,m}^{(i)}$ is an instance of the multi-variate model configuration \mathbf{X} , and $\pi_{k,m}^{(i)}$ is the corresponding particle weighting. Each annealing run can be broken down as follows (the process is illustrated in figure 2).

1. For every time step t_k an annealing run is started at layer M , with $m = M$.
2. Each layer of an annealing run is initialised by a set of un-weighted particles $\mathcal{S}_{k,m}$.
3. Each of these particles is then assigned a weight

$$\pi_{k,m}^{(i)} \propto w_m(\mathbf{Z}_k, \mathbf{s}_{k,m}^{(i)}) \quad (8)$$

which are normalised so that $\sum_N \pi_{k,m}^{(i)} = 1$. The set of weighted particles $\mathcal{S}_{k,m}^\pi$ has now been formed.

4. N particles are drawn randomly from $\mathcal{S}_{k,m}^\pi$ with replacement and with a probability equal to their weighting $\pi_{k,m}^{(i)}$. As the n^{th} particle $\mathbf{s}_{k,m}^{(n)}$ is chosen it is used to produce the particle $\mathbf{s}_{k,m-1}^{(n)}$ using

$$\mathbf{s}_{k,m-1}^{(n)} = \mathbf{s}_{k,m}^{(n)} + \mathbf{B}_m \quad (9)$$

where \mathbf{B}_m is a multi-variate gaussian random variable with variance \mathbf{P}_m and mean $\mathbf{0}$.

5. The set $\mathcal{S}_{k,m-1}$ has now been produced which can be used to initialise layer $m - 1$. The process is repeated until we arrive at the set $\mathcal{S}_{k,0}^\pi$.
6. $\mathcal{S}_{k,0}^\pi$ is used to estimate the optimal model configuration \mathcal{X}_k using

$$\mathcal{X}_k = \sum_{i=1}^N \mathbf{s}_{k,0}^{(i)} \pi_{k,0}^{(i)}. \quad (10)$$

7. The set $S_{k+1,M}$ is then produced from $S_{k,0}^\pi$ using

$$\mathbf{s}_{k+1,M}^{(n)} = \mathbf{s}_{k,0}^{(n)} + \mathbf{B}_0. \quad (11)$$

This set is then used to initialise layer M of the next annealing run at t_{k+1} .

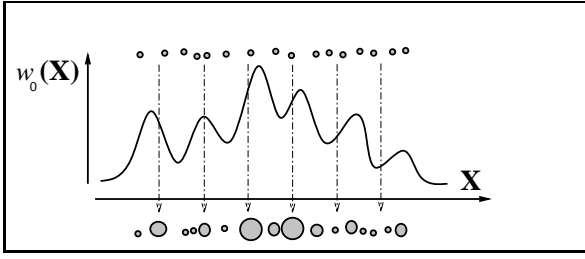


Figure 1: **Illustration of the annealed particle filter with $M = 1$.** Even though a large number of particles are used (so that an equivalent number of weighting function evaluations are made as in figure 2), the search is misdirected by local maxima. From the resulting weighted set it is very hard to tell where the global maximum of w_0 lies.

5. Setting the tracking parameters

As stated previously the function $w_m(\mathbf{Z}_k, \mathbf{X})$, used in each layer of the annealing process is determined by

$$w_m(\mathbf{Z}, \mathbf{X}) = w(\mathbf{Z}, \mathbf{X})^{\beta_m} \quad (12)$$

with $\beta_0 > \beta_1 > \dots > \beta_M$. The value of β_m will determine the rate of annealing at each layer. A large β_m will produce a peaked weighting function w_m resulting in a high rate of annealing. Small values of β_m will have the opposite effect.

If the rate of annealing is too high the influence of local maxima will distort the estimate of \mathcal{X}_k as seen in figure 1. If the rate is too low \mathcal{X}_k will not be determined with enough resolution (unless more layers are used wasting computational resources).

A good measure of the effective number of particles that will be chosen for propagation to the next layer is the survival diagnostic \mathcal{D} (taken from [14]) where

$$\mathcal{D} = \left(\sum_{n=1}^N (\pi^{(n)})^2 \right)^{-1} \quad (13)$$

and from this a good measure for the rate of annealing can be derived, called the particle survival rate α [5] [12]

$$\alpha = \frac{\mathcal{D}}{N}. \quad (14)$$

Now a measure for the rate of annealing has been derived it is possible to set the values of $\beta_0^k, \dots, \beta_M^k$ at each time step t_k . At layer m in an annealing run, β_m^{k-1} from t_{k-1} is used to calculate a preliminary set of particle weights for $S_{k,m}^\pi$. From this set an initial rate of annealing α_{init} can

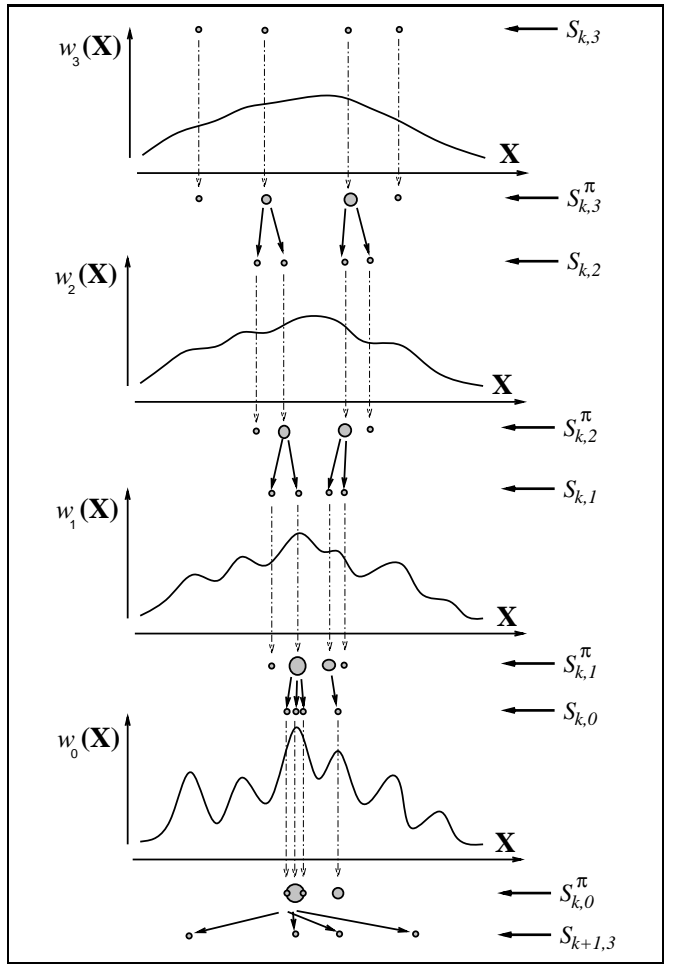


Figure 2: **Illustration of the annealed particle filter with $M = 3$.** With a multi-layered search the sparse particle set is able to gradually migrate towards the global maximum without being distracted by local maxima. The final set $S_{k,0}^\pi$ provides a good indication of the weighting function's global maximum.

be calculated using equations 13 and 14. It can be shown that $\mathcal{D}(\beta)$ is monotonic decreasing in β so that, given α , the equation

$$\mathcal{D}(\beta) = N\alpha \quad (15)$$

has a unique solution for β . With this knowledge we can minimise the error function Δ_α between the desired rate of annealing α_m and the initial rate of annealing α_{init}

$$\Delta_\alpha(\beta) = (\alpha_m - \alpha_{init}(\beta)), \quad (16)$$

using gradient descent to find the desired β_m^k . Note that this does not mean the weights have to be completely re-evaluated each time β_m^k is adjusted during gradient descent. Since $w_m(\mathbf{Z}, \mathbf{X}) = w(\mathbf{Z}, \mathbf{X})^{\beta_m}$ the values $w(\mathbf{Z}, \mathbf{X} = \mathbf{s}_{k,m}^{(i)})$, $i : 1 \dots N$ can be stored for each set $S_{k,m}^k$ and β_m^k applied to each individual weight as appropriate to produce $S_{k,m}^\pi$.

How then are the appropriate values for $\alpha_0 \dots \alpha_M$ determined? There are also a number of other tracking parameters

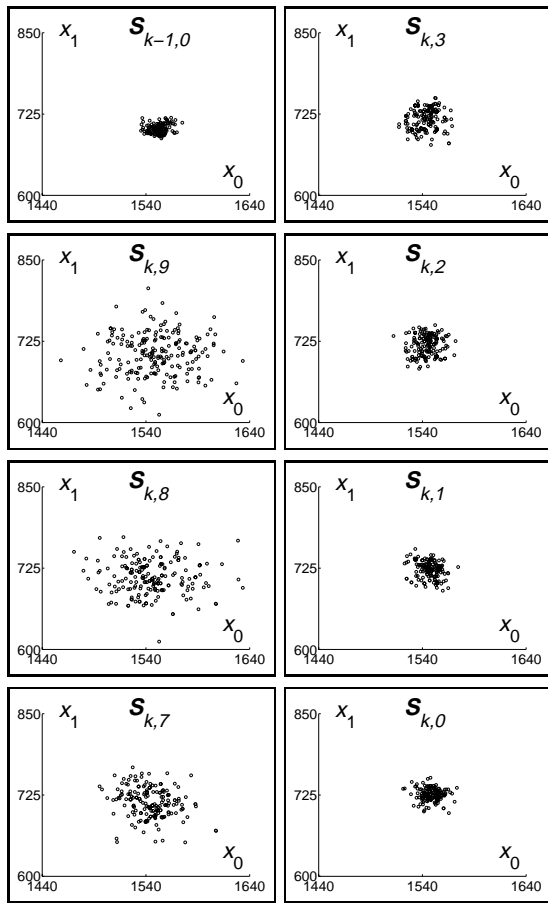


Figure 3: **Annealed particle filter in progress.** The sets $\mathcal{S}_{k,m}$ are plotted here, taken while tracking the walking person as seen in figure 9. Only the horizontal translation components x_0 and x_1 of the model configuration vector \mathbf{X} are shown. Starting with $\mathcal{S}_{k-1,0}$ from the previous time step the particles are diffused to form $\mathcal{S}_{k,9}$ which easily covers the expected range of translational movement of the subject. The particles are then slowly annealed over 10 layers (the sets $\mathcal{S}_{k,6}$ to $\mathcal{S}_{k,4}$ are omitted for brevity) to produce $\mathcal{S}_{k,0}$ which is clustered around the maximum of the weighting function.

that need to be set before tracking can begin, including the number of particles N , the number of annealing layers M and the diffusion variance vectors $\mathbf{P}_0 \dots \mathbf{P}_M$. A tentative framework has been developed to allocate values to these parameters although it is acknowledged that more work needs to be done in this area.

1. The first step is to decide on how many annealing layers are needed. It was found that doubling the number of annealing layers reduces the number of particles needed for successful tracking by more than half. This will only work up to a point however as there seems to be a minimum number (N) of particles needed for tracking no matter how many layers are used. Using a 30 DOF model it was found that setting $M = 10$ with $N \geq 200$ worked well.
2. Each element in the vector \mathbf{P}_0 is allocated a value equal to half the maximum expected movement of the corre-

sponding model configuration parameter over one time step. In this way the set $\mathcal{S}_{k+1,M}$ should cover all possible movements of the subject between time t_k and t_{k+1} . The amount of diffusion added to each successive annealing layer should decrease at the same rate as the resolution of the set $\mathcal{S}_{k,m}$ increases. It has been found that setting

$$\mathbf{P}_m = \mathbf{P}_0 (\alpha_M \alpha_{M-1} \dots \alpha_m) \quad (17)$$

produces good results.

3. The appropriate rates of annealing $\alpha_0 \dots \alpha_M$ are influenced by the number of annealing layers used. With a higher number of annealing layers a lower rate of annealing can be used to obtain the desired resolution. It was found that while using 10 annealing layers setting $\alpha_0 = \alpha_1 = \dots = \alpha_M = 0.5$ provided sufficient resolution of \mathcal{X}_k .

6. The model

The articulated model of the human body used in this paper is built around the framework of a kinematic chain, as seen in figure 4. Each limb is fleshed out using conic sections with elliptical cross-sections. It is believed that such a model has a number of advantages including computational simplicity, high-level interpretation of output and compact representation.

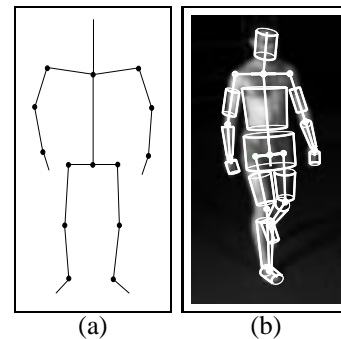


Figure 4: **The model** is based on a kinematic chain consisting of 17 segments (a). Six degrees of freedom are given to base translation and rotation. The shoulder and hip joints are treated as sockets with 3 degrees of freedom, the clavicle joints are given 2 degrees of freedom (they are not allowed to rotate about their own axis) and the remaining joints are modelled as hinges requiring only one. This results in a model with 29 degrees of freedom and a configuration vector $\mathbf{X} = \{x_1 \dots x_{29}\}$. The model is fleshed out by conical sections (b).

7. The weighting function

When deciding which image features are to be used to construct the weighting function a number of factors must be taken into account.

- *Generality.* The image features used should be invariant under a wide range of conditions so that the same tracking framework will function well in a broad variety of situations.

- *Simplicity.* In an effort to make the tracker as efficient as possible the features used must be easy to extract.

Two image features were chosen to construct the weighting function: edges and foreground silhouette. The strongest continuous edges produced by a human subject in an image usually provide a good outline of visible arms and legs and are mostly invariant to colour, clothing texture, lighting and pose. In severely cluttered environments or when the subject is wearing very baggy clothes edges may lose some of their usefulness, however in most situations they provide a good basis for a weighting function. A gradient based edge detection mask is used to detect edges. The result is thresholded to eliminate spurious edges, smoothed with a Gaussian mask and remapped between 0 and 1. This produces a pixel map (figure 5(b)) which each pixel is assigned a value related to its proximity to an edge.

A sum-squared difference (SSD) function $\Sigma^e(\mathbf{X}, \mathbf{Z})$ is then computed using

$$\Sigma^e(\mathbf{X}, \mathbf{Z}) = \frac{1}{N} \sum_{i=1}^N (1 - p_i^e(\mathbf{X}, \mathbf{Z}))^2 \quad (18)$$

where \mathbf{X} is the model's configuration vector and \mathbf{Z} is the image from which the pixel map is derived. $p_i(\mathbf{X}, \mathbf{Z})$ are the values of the edge pixel map at the N sampling points taken along the model's silhouette as seen in figure 6(a).

The second feature extraction performed on the image is foreground-background segmentation. Thresholded background subtraction was used here to separate the subject from the background and typical results can be seen in figure 5(c). This may be inappropriate in some environments with a lot of background movement where more sophisticated methods may have to be employed. Most foreground segmentation techniques are largely invariant to clothing, lighting, pose motion and environment and as such provide an excellent image feature for a general human motion capture system. Once again a pixel map is constructed, this time with foreground pixels set to 1 and background to 0 (figure 5(b)), and an SSD is computed

$$\Sigma^r(\mathbf{X}, \mathbf{Z}) = \frac{1}{N} \sum_{i=1}^N (1 - p_i^r(\mathbf{X}, \mathbf{Z}))^2 \quad (19)$$

where $p_i(\mathbf{X}, \mathbf{Z})$ are the values of the foreground pixel map at the N sampling points taken from the interior of the conical sections as seen in figure 6(b).

To combine the edge and region measurements the two SSD's are added together and the result exponentiated to give

$$w(\mathbf{X}, \mathbf{Z}) = \exp - (\Sigma^e(\mathbf{X}, \mathbf{Z}) + \Sigma^r(\mathbf{X}, \mathbf{Z})) . \quad (20)$$

When there is more than one camera the measurements are combined in a similar way, giving

$$w(\mathbf{X}, \mathbf{Z}) = \exp - \left(\sum_{i=1}^C (\Sigma_i^e(\mathbf{X}, \mathbf{Z}) + \Sigma_i^r(\mathbf{X}, \mathbf{Z})) \right) \quad (21)$$

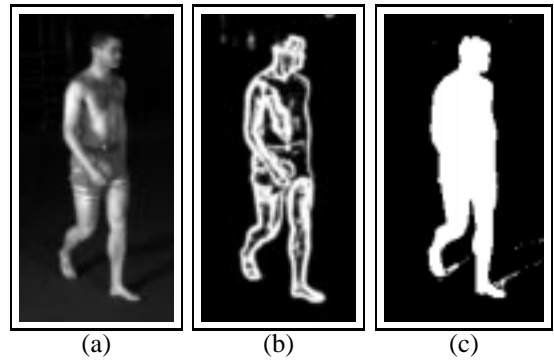


Figure 5: **Feature extraction.** A gradient based edge detection mask is used to find edges. The result is thresholded to eliminate spurious edges and smoothed using a Gaussian mask to produce a pixel map (b) in which the value of each pixel is related to its proximity to an edge. The foreground is segmented using thresholded background subtraction to produce the pixel map (c) used in the weighting function.

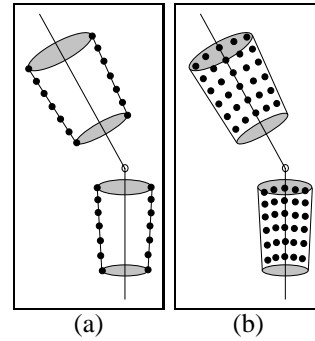


Figure 6: **Configurations of the pixel map sampling points** $p_i(\mathbf{X}, \mathbf{Z})$ for the edge based measurements (a) and the foreground segmentation measurements (b). The sampling points for the edge measurements are located along the occluding contours of the model's conical sections that have been projected into the image. The sampling points for the foreground segmentation measurements are taken from a grid within these occluding contours.

where C is the number of cameras and $\sum_i^*(\mathbf{X})$ is from camera i . An example of the output of this weighting function can be seen in figure 7.

8. Results

The human motion capture system was used to track a subject performing two different activities. The first was walking in a circle as seen in figure 9. The second was the subject stepping over a box, turning around and stepping over it again as seen in figure 10.

Three cameras were used to capture the motion and all three views can be seen in the corresponding figures. The same tracking parameters were used in all three sequences, which demonstrate the tracker's ability to follow a wide range of human movement. A comparison of the annealed particle filter with standard Condensation can be seen in figure 8 showing the improved tracking performance given

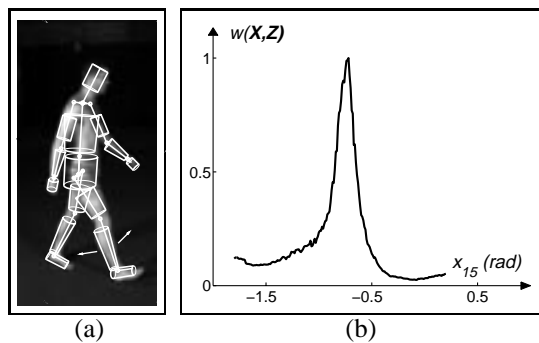


Figure 7: **Example output of the weighting function** obtained by varying only component x_{15} of \mathbf{X} (the right knee angle) using the image and model configuration seen in (a). The function is highly peaked around the correct angle of -0.7 radians (b).

equivalent computational resources.

The tracker was run on an SGI Octane with a single 175 MHz R10000 CPU. Using 10 annealing layers with 200 particles the system took approximately 1 hour to process 5 seconds of footage. The tracker is however still in the prototype stage and we hope to achieve near to real-time performance in the future.

9. Conclusion

The main obstacle to practical human motion capture is the high number of dimensions associated with an articulated full-body model. Algorithms, either deterministic or stochastic, that search such a space without constraint, fall foul of exponential growth in computational complexity. One solution to this problem is to introduce constraints — either labelling using markers or colour coding, prior assumptions about motion trajectories or view restrictions. Using these constraints limits the generality of the resulting tracker. In developing an unconstrained full-body tracker we have had to address the problem of tracking in high dimensional configuration spaces.

Although the Condensation algorithm has been shown to be a robust and powerful stochastic filter it is not feasible to apply it in high dimensional configuration spaces. As a consequence we developed a modified particle filter. It uses a continuation principle, based on annealing, to gradually introduce the influence of narrow peaks in the fitness function. The new algorithm, termed *annealed particle filtering*, is capable of recovering full articulated body motion efficiently. As can be seen in the practical results the new method leads to very robust tracking even when faced with complex and difficult sequences of movement. Since the number of particles required for successful tracking was reduced by over a factor of 10 a considerable step towards real-time tracking has been made.

Acknowledgements This work was supported by Oxford Metrics and EPSRC grant GR/M15262 (JD, IR). IR acknowledges the support of an EPSRC Advanced Research

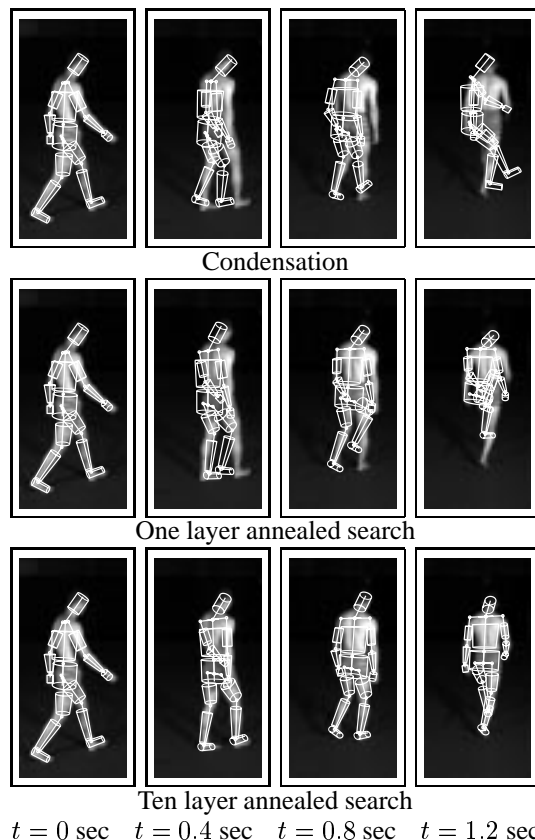
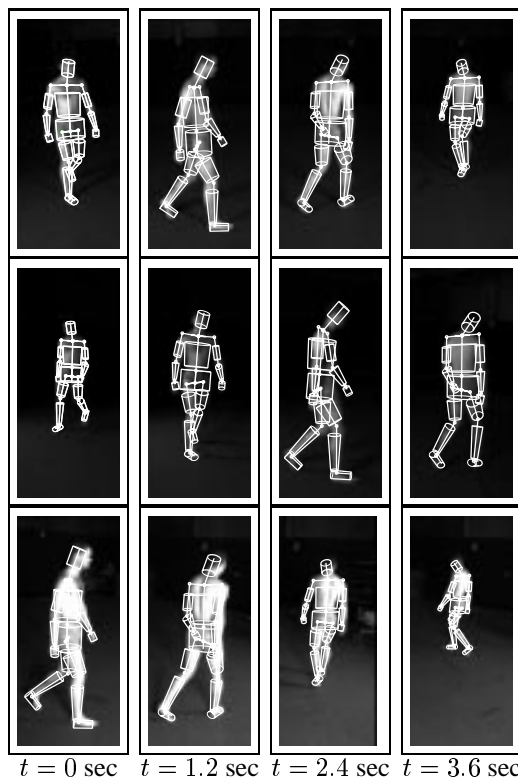


Figure 8: **A comparison of Condensation with the annealed particle filter.** At top the results of tracking with 4000 particles using standard Condensation can be seen. An observation model based on a fusion of edge information (as in [1]) and SSD template correlation was used. Tracking gradually deteriorates until terminal failure after 1.2 seconds. Experiments with 40000 particles were carried out taking over 30 hours to process just 4 seconds of video, still with negative results. An annealed search using 4000 particles with one layer fails little better (middle), also suffering terminal failure after 1.2 seconds. The main difference between normal Condensation and single layer annealed sampling is that instead of a complicated observation model, a simple weighting function is used as described in section 7. This results in a 4 fold speed increase. An annealed search using 400 particles and 10 layers (ie. 4000 weighting function evaluations per frame) tracks very well. It was found in practice that good results could be achieved with as little as 100 particles.

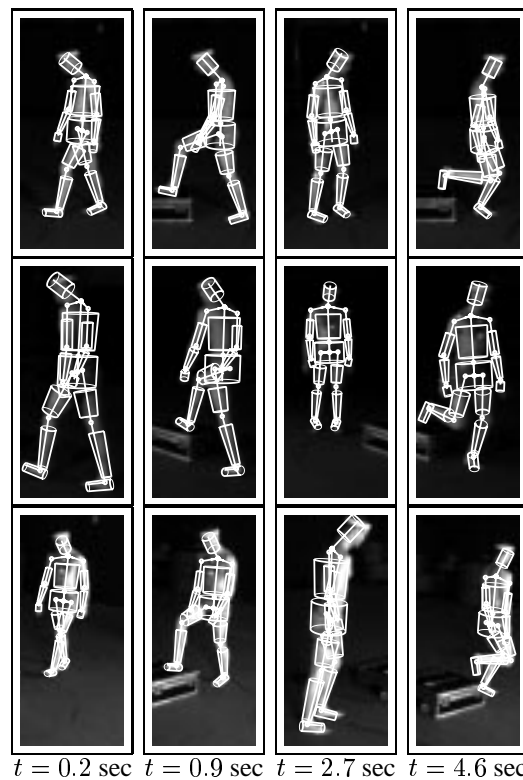
Fellowship. JD would also like to thank Ben North and the rest of the Oxford Visual Dynamics group for all their help over the last year.

References

- [1] Blake, A., and Isard, M. *Active contours*. Springer, 1998.
- [2] Bregler, C., and Malik, J. Tracking people with twists and exponential maps. In *Proc. CVPR* (1998).
- [3] Deutscher, J., Blake, A., North, B., and Basclé, B. Tracking through singularities and discontinuities by random sampling. In *Proc. 7th Int. Conf. on Computer Vision* (1999), vol. 2, 1144–1149.



$t = 0 \text{ sec}$ $t = 1.2 \text{ sec}$ $t = 2.4 \text{ sec}$ $t = 3.6 \text{ sec}$



$t = 0.2 \text{ sec}$ $t = 0.9 \text{ sec}$ $t = 2.7 \text{ sec}$ $t = 4.6 \text{ sec}$

Figure 9: **Walking in a circle.** Using three cameras (arrayed here from top to bottom) a person is tracked over 4 seconds while walking in a circle. The tracker maintains an accurate lock throughout. 10 annealing layers were used with 200 particles for this sequence. Download the movie from www.robots.ox.ac.uk/~jdeutsch/HMC/.

Figure 10: **Stepping over a box.** Using three cameras (arrayed here from top to bottom) a person is tracked over 5 seconds while stepping over a box, turning around and stepping over the box again. The tracker maintains an accurate lock throughout. 10 annealing layers were used with 200 particles for this sequence. Download the movie from www.robots.ox.ac.uk/~jdeutsch/HMC/.

[4] Gavrilu, D., and Davis, L. 3d model-based tracking of humans in action: a multi-view approach. *Proc. Conf. Computer Vision and Pattern Recognition* (1996), 73–80.

[5] Geweke, J. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* 57 (1989), 1317–1339.

[6] Goncalves, L., di Bernardo, E., Ursella, E., and Perona, P. Monocular tracking of the human arm in 3D. In *Proc. 5th Int. Conf. on Computer Vision* (1995), 764–770.

[7] Haritaoglu, I., Harwood, D., and Davis, L. w^4s : A real-time system for detecting and tracking people in 2.5D. In *Proc. 5th European Conf. Computer Vision* (Freiburg, Germany, June 1998), vol. 1, Springer Verlag, 877–892.

[8] Hogg, D. Model-based vision: a program to see a walking person. *J. Image and Vision Computing* 1, 1 (1983), 5–20.

[9] Isard, M., and Blake, A. Visual tracking by stochastic propagation of conditional density. In *Proc. 4th European Conf. Computer Vision* (Cambridge, England, Apr 1996), 343–356.

[10] Ju, S., Black, M., and Yacoob, Y. Cardboard people: A parameterized model of articulated motion. In *2nd Int. Conf. on Automatic Face and Gesture Recognition, Killington, Vermont* (1996), 38–44.

[11] Kirkpatrick, S., Gellatt, C., and Vecchi, M. Optimisation by simulated annealing. Tech. rep., IBM Thomas J. Watson Research Centre, Yorktown Heights, NY, USA, 1982.

[12] MacCormick, J. *Probabilistic models and stochastic algorithms for visual tracking*. PhD thesis, University of Oxford, 2000.

[13] MacCormick, J., and Blake, A. A probabilistic exclusion principle for tracking multiple objects. In *Proc. 7th Int. Conf. on Computer Vision* (1999), vol. 1, 572–578.

[14] MacCormick, J., and Blake, A. Partitioned sampling, articulated objects and interface-quality hand tracking. In *Accepted to ECCV 2000* (2000).

[15] Niyogi, S., and Adelson, E. Analysing and recognising walking figures in xyt. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition* (1994), 469–474.

[16] Rehg, J., and Morris, D. Singularities in articulated object tracking with 2-d and 3-d models. Tech. rep., Digital Equipment Corporation, Cambridge Research Lab, 1997.

[17] Rohr, K. Human movement analysis based on explicit motion models. In *Motion-Based Recognition*. Kluwer Academic Publishers, Dordrecht Boston, 1997, ch. 8, 171–198.

[18] Sullivan, J., Blake, A., Isard, M., and MacCormick, J. Object localization by bayesian correlation. In *Proc. 7th Int. Conf. on Computer Vision* (1999), vol. 2, 1068–1075.

[19] Vicon web based literature. URL <http://www.metrics.co.uk>, 1999. valid December 1999.