International Journal of Computer Vision 61(2), 185–205, 2005

Articulated Body Motion Capture by Stochastic Search

JONATHAN DEUTSCHER AND IAN REID
Department of Engineering Science, University of Oxford, Oxford, OX13PJ, United Kingdom
jdeutsch@robots.ox.ac.uk
ian.reid@eng.ox.ac.uk

Received August 19, 2003; Revised April 9, 2004; Accepted April 9, 2004

First online version published in October, 2004

Abstract. We develop a modified particle filter which is shown to be effective at searching the high-dimensional configuration spaces (c. 30 + dimensions) encountered in visual tracking of articulated body motion. The algorithm 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. A mechanism for achieving a soft partitioning of the search space is described and implemented, and shown to improve the algorithm's performance. Likewise, the introduction of a crossover operator is shown to improve the effectiveness of the search for kinematic trees (such as a human body). Results are given for a variety of agile motions such as walking, running and jumping.

Keywords: human motion capture, visual tracking, particle filtering, genetic algorithms

1. Introduction

A popular form of motion capture, for tasks such as gait analysis and computer animation, involves attaching a number of retro-reflective markers to a subject's body and viewing the motion of the markers over time using a set of calibrated cameras. The easily-recovered image positions of the markers are transformed into 3D trajectories via triangulation of the measurements, and a parameterised representation of the subject's movements can be calculated.

The use of markers is intrusive and restrictive, and necessitates the use of potentially expensive, specialised capture hardware. The goal of markerless motion capture is to reproduce the performance of marker-based methods in a system using conventional cameras and without the use of special apparel or other equipment. For this reason recent years have seen a huge growth in research in the computer vision community with the aim of recovering motion data directly from images, without markers. However, full-body tracking from standard images is a challenging problem, and markerless system presented to date rarely achieve the following combination of capabilities of current marker-based systems: full 3D motion recovery; robust tracking of rapid, arbitrary movement; high accuracy; easy application to new scenarios; on-line model acquisition; real-time, or near real-time processing.

A major problem which confronts all attempts to satisfy these criteria is the high dimensionality of the configuration space, and the exponentially increasing computational cost that results. A realistic articulated model (see Fig. 4) of the human body usually has at least 25 DOF. The model used in this paper for example has between 29 and 34 DOF, and models employed for commercial character animation usually have over 40.

In this paper we describe a multi-camera system for markerless human motion capture which goes some way to achieving the goals above. The work described combines and extends our previous efforts published

P1: xxx
International Journal of Computer Vision KL3179-05/5384380 September 29, 2004 20:18

in short form in Deutscher et al. (2000, 2001). Our approach is characterised by the following: 1. articulated body model, 2. weak dynamical modelling, 3. edge and background subtraction image measurements, and 4. a particle-based stochastic search algorithm. The key contributions comprise:

- The development of a novel, particle-based stochastic search algorithm, called annealed particle filtering. The method uses a continuation principle, based on annealing, to introduce the influence of narrow peaks in the fitness function gradually. This is introduced in Section 3.
- The annealed particle filter is applied to the problem of markerless human motion capture, and shown to be more effective and efficient than, for example, Condensation (Blake and Isard, 1998), at localising the pose. Section 4 discusses implementation issues and Section 5 shows results.
- We demonstrate how adaptive selection of the variances/covariances which control the diffusion during annealing can lead to what can be thought of as a "soft" hierarchical partitioning of the configuration space, and hence to further gains in efficiency.
- We introduce a crossover operator, analogous that that found in Genetic Algorithms, into the particle filtering framework. We demonstrate that this operator improves the ability of the algorithm to search the configuration spaces of objects whose articulations can be modelled as a kinematic tree. In particular we show results for reliable and efficient tracking for walking, running and jumping with no special training of the dynamics of any activity.

  These latter two developments are discussed in Section 6.

We begin with a review of relevant literature in Section 2, including a detailed discussion of the two most closely associated technologies: particle filtering and simulated annealing.

2. Background

2.1. *Visual Tracking and Particle Filters*

Full-body motion capture is an example of model-based tracking; i.e. the process of sequentially estimating the parameters of a model of a target over time from visual data. Typically a priori knowledge about the target's observable properties (such as its geometry) are compared with the visual data from an image stream, to estimate a best fit for each frame in the scene. Thus the principal components usually comprise a target model, an image search method, and a dynamical model.

The system of Plänkers and Fua (2003) represents one of the best examples of this paradigm in its simplest form. Their system does not have a (strong) model of the person's dynamics (in contrast to Sidenbladh et al. (2000), e.g., see below) or have a sophisticated multimodal search algorithm such as we describe. Rather, the key to the success of their system is in much more careful modelling of the shape and appearance than in most other work, and in the use of binocular disparity maps as well as silhouette data. Unlike most other work (including our own) their system estimates the size of the body as well as the pose parameters.

In considering the role of the other two system components, search and dynamics it is useful to discuss the influential work of Harris (1992). He showed how rapid motions can be tracked by constraining the search area via a predicted motion of the object. Harris used rigid polyhedral models (and simple surfaces of revolution) and sought the 6 DOF pose of object. Given a predicted location, the system searches from predicted edge locations along 1D profiles to find "actual" edges. These 1D measurements are then combined to obtain a pose update.

Drummond and Cipolla (2001) showed how many of the ideas in Harris' system can be extended to articulated objects by effectively tracking body parts using Harris' method but enforcing global consistency via kinematic constraints.

A second and arguably more important innovation in Harris (1992) was to place the tracking system within the framework of a Kalman Filter, a provably optimal recursive estimator for linear systems which can be thought of as an algorithm for sequential propagation of Gaussian probability densities.

A natural step would be to consider the use of a Kalman Filter, (or its extension to non-linear measurements and/or dynamics, the Extended Kalman Filter or EKF) for articulated body tracking. Wachter and Nagel (1999) demonstrated this for single view tracking using image motion and edges (though the results show only motion parallel to the image plane). More recently Mikic et al. (2001) demonstrated the extraction and filtering of pose parameters from a volumetric model obtained by "carving" space using silhouettes from multiple cameras.

P1: xxx
International Journal of Computer Vision KL3179-05/5384380 September 29, 2004 20:18

While Gaussian uncertainty is sufficient for modelling many motion and measurement noise sources, the Kalman Filter has been shown to fail catastrophically in cases where the true probability function has a very different shape. In particular attempts to track objects moving against a very cluttered background, where measurement densities include the chance of detecting erroneous image features and are therefore multi-modal, lead to tracking failure for Harris' algorithm and many of its ilk.

An alternative, more general approach is particle filtering, in which arbitrary densities are approximated. This was introduced in the context of visual tracking in the form of the Condensation algorithm (Isard and Blake, 1996). A posterior density $p(\mathbf{X}\,|\,\underline{\mathbf{Z}}_{\mathbf{t}})$ representing current knowledge about the model state $\mathbf{X}$ after incorporation of all measurements up to the current time-step t, $\underline{\mathbf{Z}}_{\mathbf{t}}$, is represented by a finite set of *normalised weighted particles*, or samples,

$$\left\{\left(\mathbf{s}_t^{(0)}, \pi_t^{(0)}\right)\cdots\left(\mathbf{s}_t^{(N)}, \pi_t^{(N)}\right)\right\}.$$

An estimate of the state $\mathcal{X}_t$ at each time-step t can easily be estimated by the sample mean of the posterior density, $p(\mathbf{X}\,|\,\underline{\mathbf{Z}}_{\mathbf{t}})$,

$$\mathcal{X}_t = \mathcal{E}[\mathbf{X}] = \sum_{n=1}^{N} \pi_t^{(n)} \mathbf{s}_t^{(n)} \tag{1}$$

or the mode

$$\mathcal{X}_t = \mathcal{M}[\mathbf{X}] = \mathbf{s}_t^{(j)}, \quad \pi_t^{(j)} = \max\left(\pi_t^{(n)}\right). \tag{2}$$

Essentially, a smooth probability density function is approximated by a finite collection of weighted sample points, and it can be shown that as the number of points tends to infinity the behaviour of the particle set is indistinguishable from that of the smooth function. Tracking with a particle filter works by:

1. Resampling, in which a weighted particle set is transformed into a set of evenly weighted particles distributed with concentration dependent on probability density;
2. Stochastic movement and dispersion of the particle set in accordance with a motion model to represent the growth of uncertainty during movement of the tracked object;
3. Measurement, in which the likelihood function is evaluated at each particle site, producing a new weight for each particle proportional to how well it fits image data. The weighted particle set produced represents the new (posterior) probability density after movement and measurement.

Particle filtering works well for tracking in clutter because it can represent arbitrary functional shapes and propagate multiple hypotheses. Less likely model configurations will not be thrown away immediately but given a chance to prove themselves later on, resulting in more robust tracking. In its original implementation, Condensation demonstrated robust tracking in low-dimensional configuration spaces (up to about 10 DOF) in the presence of significant clutter. Even in the absence of a cluttered background, the complicated nature of the observation process during human motion capture causes the posterior density to be non-Gaussian and multi-modal as shown experimentally by Deutscher et al (1999). Condensation has indeed been implemented successfully for short human motion capture sequences (see Deutscher et al. (2000) and Sidenbladh et al. (2000)), however, in the high-dimensional configuration spaces occurring in human motion capture and other domains, there are serious problems with Condensation arising from the inability of a manageable size particle set (of, say, a few thousand particles), adequately to populate the space and represent an arbitrary density. In fact it has been shown by MacCormick and Isard (2000) that $N \geq \frac{\mathcal{D}_{\min}}{\alpha^d}$ 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 $\alpha \ll 1$. Clearly when d is large normal particle filtering becomes infeasible.

Cham and Rehg (1999) proposed the use of a multiple hypothesis tracker which represented the posterior distribution as a piecewise Gaussian. As only local modes are propagated between frames, the solution is computationally much cheaper than Condensation, but they avoid the pitfalls of a single hypothesis tracker. Unlike our work, in which we explicitly model the joint angles and overall pose degrees of freedom, they use a so-called scaled prismatic model which explicitly models 2D in-plane translation and rotation, but models out of plane rotation via a per-link independent scaling.

Partitioned sampling was developed by MacCormick and Blake (1999) as a variation on Condensation to reduce the number of particles needed to track more than one object, and applied by MacCormick and Isard (2000) to the problem of tracking articulated objects. Using partitioned sampling reduces the number of particles required to $N >= \frac{\mathcal{D}_{\min}}{\alpha}$ making the problem tractable. However, this assumes that the configuration space can be sliced so that one can construct an observation density $p(\mathbf{Z}_t \mid x_i)$ for each dimension x_i of the model configuration vector $\mathbf{X} = \{x_0 \ldots x_d\}$. This assumption, that it is possible independently to localise separate parts of an articulated model, is similar to that made by Gavrila and Davis (1996) to enable a hierarchical search.

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. (1999). Called *layered sampling* it is centred around the concept of importance resampling. The technique we present in this paper bears some similarity to layered sampling, but experimental evidence suggests our technique is more effective at reducing the number of particles required when the dimensionality of the search space approaches 30.

Two successful recent approaches which use particle filtering are due to Sminchisescu and Triggs (2001) and Sidenbladh et al. (2000). Both are concerned with monocular tracking (in some important ways more difficult than the multi-camera case) but in other respects problem is essentially the same: how can a high dimensional space be adequately populated with a particle set of manageable size? Their approaches to this problem are quite different and in some ways complementary.

The former introduces the idea of *covariance sampling*, spreading particles in areas where there is least confidence in the localisation. This idea is very closely related to our soft partitioning approach developed in Section 6.1. More recently they have extended this work explicitly to take into account the particular ambiguities that arise from human kinematics, "scattering" particles into areas of potential ambiguity and therefore making better use of the particle set Sminchisescu and Triggs (2003).

The latter work (Sidenbladh et al., 2000, 2002) on the other hand, takes the approach that dynamical modelling can be used to obtain strong, predictive priors, reducing the search space to manageable proportions. Indeed in Sidenbladh et al. (2000) tracking was restricted, via the learnt dynamics, to the case of walking people. More recently however (Sidenbladh et al., 2002) showed how a database of motions could be constructed and efficiently indexed in order to obtain predictions over a wide class of motions.

In addition to the problems of representing PDFs via particle sets in high dimensional spaces, a second difficulty is associated with constructing a valid observation model $p(\mathbf{Z}_t \mid \mathbf{X})$ as a normalised probability density distribution. Even if such a likelihood model can be constructed the cost of evaluating it can be prohibitive.[1] Often an intuitive weighting function $w(\mathbf{Z}_t, \mathbf{X})$ can be constructed that approximates the probabilistic likelihood $p(\mathbf{Z}_t \mid \mathbf{X})$ but which requires much less computational effort to evaluate.

In this paper we reduce the problem from propagating the conditional density $p(\mathbf{X} \mid \underline{\mathbf{Z}_t})$ using $p(\mathbf{Z} \mid \mathbf{X})$, simply to finding the configuration $\mathcal{X}_t$ which returns the maximum value from a simple and efficient weighting function $w(\mathbf{Z}_t, \mathbf{X})$ at each time t, given $\mathcal{X}_{t-1}$. By doing this gains are made on two fronts: (i) it is possible to make do with fewer likelihood (or weighting function) evaluations because the function $p(\mathbf{X} \mid \mathbf{Z}_t)$ no longer has to be fully represented; and (ii) an evaluation of a simple weighting function $w(\mathbf{Z}_t, \mathbf{X})$ requires less computational effort when compared to an evaluation of the observation model $p(\mathbf{Z}_t \mid \mathbf{X})$. The main disadvantage is that we no longer work within a truly Bayesian framework.

We retain the use of 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. In order most effectively to optimise the non-convex weighting function we use an approach similar to that of simulated annealing.

2.2. Simulated Annealing

The Markov chain based method of simulated annealing was proposed by Kirkpatrick et al. (1983) as a means to optimise a multi-modal objective function $U(\mathbf{x})$. It proceeds by defining a distribution over the function values

$$P(\mathbf{x}) = \text{const } e^{-\lambda U(\mathbf{x})}$$

The aim is then to generate samples $\mathbf{x}_i$ from this distribution, in the knowledge that as $\lambda \to \infty$, the probability mass concentrates on the minumum of U, and hence the samples $\mathbf{x}_i$ will cluster around the minimum value state.

Samples from the distribution are generated in a straightforward fashion using the Metropolis-Hastings algorithm (Metropolis et al., 1953) which generates a Markov sequence of points whose distribution will converge to P: a new candidate point $\mathbf{x}'$ in a sequence is generated "at random", and accepted with probability:

$$\min\left(1, \frac{P(\mathbf{x}')}{P(\mathbf{x})}\right)$$

i.e. the candidate point is accepted if it improves U or with probability $e^{-\lambda[U(\mathbf{x})-U(\mathbf{x}')]}$.

Simply using a large value of λ and generating a sequence starting at a random $\mathbf{x}_0$ yields poor results if U has isolated minima since the sequence can easily become trapped in a local mode of P (e.g. the closest to $\mathbf{x}_0$).

The annealing process is a heuristic for avoiding this. The initial value of λ is set to be small (or in more physical language, the temperature, which is inversely proportional to λ, is initially high). This results in a broad distribution P and hence allows free exploration of the search space. Samples are generated from this distribution, and then the value of λ increased. Samples are then generated from the new distribution starting from the final state of the previous sequence, and so on. Each increase of λ successively excludes (in a probabistic sense) regions that contain little of the probability mass of the distribution.

The set of values for $\lambda = \lambda_M \ldots \lambda_0$ is known as the annealing schedule. This schedule needs to be designed as a compromise between speed and efficacy: slow annealing is more likely to find a globally optimal solution, but is also prohibitively expensive.

The similarity with particle-based methods arises when we view this process one of generating samples from a sequence of distributions, $P_{\lambda_M} \ldots P_{\lambda_0}$, where $P_{\lambda_m}(\mathbf{x}) \propto P_{\lambda_0}(\mathbf{x})^{\beta_m}$, for $1 = \beta_0 > \beta_1 > \cdots > \beta_M$, and where $\beta_m = \lambda_m/\lambda_0$ (as described by Neal (2001) whose algorithm ours resembles). Moreover the algorithm exhibits exactly the kind of behaviour needed for the our purposes: one wants to move towards the global maximum of the weighting function $w(\mathbf{Z}_t, \mathbf{X})$, using the overall trend of the matching function as a guide, without becoming misguided by local maxima as seen in Fig. 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.

3. Annealed Particle Filter

A series of weighting functions $w_0(\mathbf{Z}, \mathbf{X})$ to $w_M(\mathbf{Z}, \mathbf{X})$ is employed in which each w_m differs only slightly

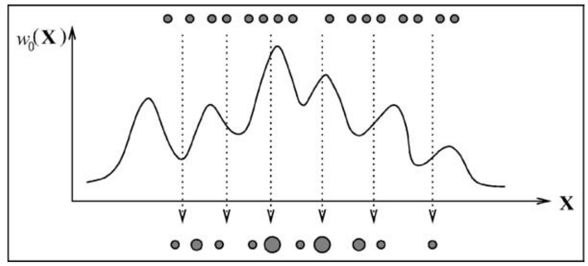


Figure 1. Illustration of the annealed particle filter with M = 1. Even though a large number of particles are used (so that an equivalent number of weighting function evaluations are made as in Fig. 2), the search is misdirected by local maxima. From the resulting weighted set it is very hard to tell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

from w_{m-1} (see Fig. 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

$$w_m(\mathbf{Z}, \mathbf{X}) = w(\mathbf{Z}, \mathbf{X})^{\beta_m}, \tag{3}$$

for $\beta_0 > \beta_1 > \cdots > \beta_M$, where $w(\mathbf{Z}, \mathbf{X})$ is the original weighting function, as suggested by the discussion in Section 2.2. 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 $\beta_0 = 1$.

A large β_m produces 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 $\mathcal{X}_t$ as seen in Fig. 1. If the rate is too low $\mathcal{X}_t$ will not be determined with enough resolution (unless more layers are used wasting computational resources). The manner in which the rate of annealing is influenced by the sequence $\beta_M, \ldots \beta_0$ is discussed in Section 3.1.

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

$$\mathcal{S}^{\pi}_{t,m} = \left\{\left(\mathbf{s}^{(0)}_{t,m}, \pi^{(0)}_{t,m}\right) \cdots \left(\mathbf{s}^{(N)}_{t,m}, \pi^{(N)}_{t,m}\right)\right\}. \tag{4}$$

An unweighted set of particles will be denoted

$$\mathcal{S}_{t,m} = \left\{\left(\mathbf{s}^{(0)}_{t,m}\right) \cdots \left(\mathbf{s}^{(N)}_{t,m}\right)\right\}. \tag{5}$$

Each particle in the set $\mathcal{S}^{\pi}_{t,m}$ is considered as an $(\mathbf{s}^{(i)}_{t,m}, \pi^{(i)}_{t,m})$ pair in which $\mathbf{s}^{(i)}_{t,m}$ is an instance of the

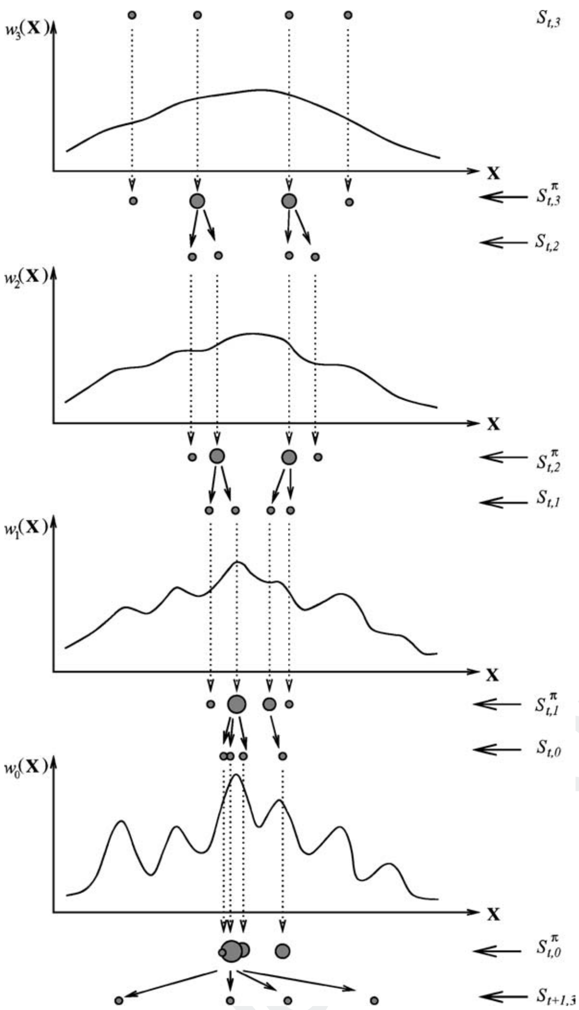
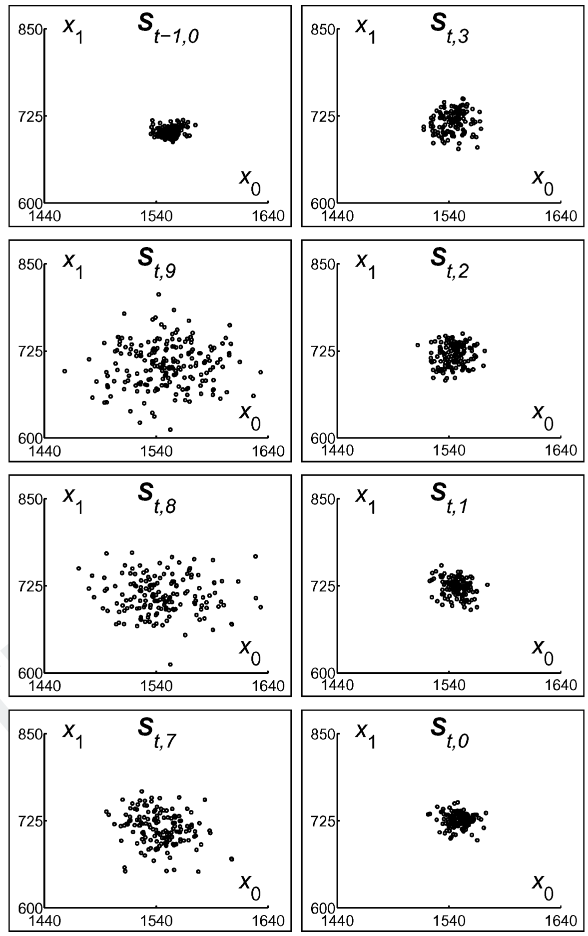


Figure 2. Illustration of the annealed particle filter with M = 3. With a multi-layered search the sparse particle set is able to migrate gradually towards the global maximum without being distracted by local maxima. The final set $\mathcal{S}^{\pi}_{t,0}$ provides a good indication of the weighting function's global maximum.



Au: Pls. cite Fig. 3 in text.

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

multi-variate model configuration **X**, and $\pi^{(i)}_{t,m}$ is the corresponding particle weighting. Each annealing run can be broken down as follows (the process is illustrated in Fig. 2).

1. For every time step t 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 $\mathcal{S}_{t,m}$.
3. Each of these particles is then assigned a weight

$$\pi^{(i)}_{t,m} \propto w_m\left(\mathbf{Z}_t, \mathbf{s}^{(i)}_{t,m}\right) \tag{6}$$

   which are normalised so that $\sum_N \pi^{(i)}_{t,m} = 1$. The set of weighted particles $\mathcal{S}^{\pi}_{t,m}$ has now been formed.
4. N particles are drawn randomly from $\mathcal{S}^{\pi}_{t,m}$ with replacement and with a probability equal to their weighting $\pi^{(i)}_{t,m}$. As the nth particle $\mathbf{s}^{(n)}_{t,m}$ is chosen it is used to produce the particle $\mathbf{s}^{(n)}_{t,m-1}$ using

$$\mathbf{s}^{(n)}_{t,m-1} = \mathbf{s}^{(n)}_{t,m} + \mathbf{B}_m \tag{7}$$

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

P1: xxx
International Journal of Computer Vision KL3179-05/5384380 September 29, 2004 20:18

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

$$\mathcal{X}_t = \sum_{i=1}^{N} \mathbf{s}^{(i)}_{t,0}\pi^{(i)}_{t,0}. \tag{8}$$

7. The set $\mathcal{S}_{t+1,M}$ is then produced from $\mathcal{S}^{\pi}_{t,0}$ using

$$\mathbf{s}^{(n)}_{t+1,M} = \mathbf{s}^{(n)}_{t,0} + \mathbf{B}_0. \tag{9}$$

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

Note that Step 7, where the particle set for the next time-step is generated, incorporates no dynamic model. There is nothing in the algorithm that precludes the use of dynamics: simply replace Eq. (9) with the more general

$$\mathbf{s}^{(n)}_{t+1,M} = f\left(\mathbf{s}^{(n)}_{t,0}\right) + \mathbf{B}_0 \tag{10}$$

where the function f represents the dynamical model. We have not done so since our focus is on tracking previously unseen/unmodelled agile motions. While a dynamical model is certainly beneficial during "steady state" tracking, it can be a hindrance if the model is poor (as it often is for agile motions). The price we pay for this is a less economical use of particles than would be ideal, and the potential for jittery trajectories. The latter could be addressed by smoothing the recovered pose/joint trajectories.

3.1. Setting the Tracking Parameters

It remains to consider how best to set the free parameters of the algorithm, and in particular, to consider how to influence the annealing schedule. In Eq. (3) it is the value of β^k_m that determines the rate of annealing at each layer.

To see how and why this is so, first note that the equivalent of temperature in our particle-based framework is the particle survival rate: the ratio of effective particles to total number of particles. If the probability mass is all concentrated in a few particles then the number of effective particles is low, and conversely, an even distribution of probability mass amongst particles signals a large number of effective particles. A sensible 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 MacCormick (2000)) where

$$\mathcal{D} = \left(\sum_{n=1}^{N}\left(\pi^{(n)}\right)^2\right)^{-1} \tag{11}$$

and from this the particle survival rate α can be estimated MacCormick (2000)

$$\alpha = \frac{\mathcal{D}}{N}. \tag{12}$$

In the case of traditional annealing, the temperature acts like a barrier, restricting the movement of samples: the cooler the temperature, the fewer the number of samples with a low function value $U(\mathbf{x})$ (energy) that will be generated. In the context of a particle set, a high survival rate corresponds to an even spread probability mass, while a low one suggests the mass is concentrated in a few particles. Hence decreasing the survival rate has the same effect as cooling the temperature in traditional annealing.

Now $\mathcal{D}$ is clearly a monotonic decreasing function of β^k_m. At a given layer, we therefore adjust the value of β^k_m to change the value of $\mathcal{D}(\beta^k_m)$ so that $\alpha = \mathcal{D}/N$ approaches a desired value. This is trivially done by searching over β^k_m (using the value from the previous time step β^{k-1}_m as the starting point) to find the value that solves the equation

$$\alpha_{\text{desired}} = \mathcal{D}\left(\beta^k_m\right)/N$$

i.e. produces the desired rate of annealing.

Note that this does not mean the weights have to be completely re-evaluated each time β^k_m is adjusted during the search. Since $w_m(\mathbf{Z},\mathbf{X}) = w(\mathbf{Z},\mathbf{X})^{\beta_m}$ the values $w(\mathbf{Z},\mathbf{X} = \mathbf{s}^{(i)}_{t,m}), i : 1\ldots N$ can be stored for each set $\mathcal{S}^{k,m}$ and β^k_m applied to each individual weight as appropriate to produce $\mathcal{S}^{\pi}_{t,m}$.

How then are the appropriate values for $\alpha_0 \ldots \alpha_M$ determined? There are also a number of other tracking parameters that need to be set before tracking can begin, including the number of particles N, the number of annealing layers M and the diffusion covariance matrices $\mathbf{P}_M \ldots \mathbf{P}_0$. 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 diagonal element in $\mathbf{P}_0$ is allocated a value equal to half the maximum expected movement of the corresponding model configuration parameter over one time step. In this way the set $\mathcal{S}_{t+1,M}$ should cover all possible movements of the subject between time t and $t+1$. The amount of diffusion added to each successive annealing layer should decrease at the same rate as the resolution of the set $\mathcal{S}_{t,m}$ increases. Our early experiments used

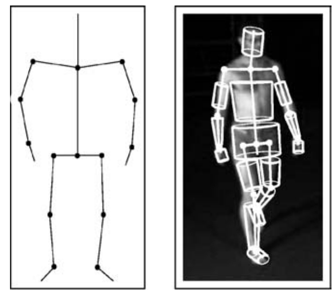
$$\mathbf{P}_m = \alpha_M \times \cdots \times \alpha_{m-1} \times \mathbf{P}_0 \qquad (13)$$

   and produced decent results, but a better, adaptive method for setting the $\mathbf{P}$ is described in Section 6.1.
3. The appropriate rates of annealing $\alpha_M \ldots \alpha_1$ 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 $\alpha_M = \ldots = \alpha_1 = 0.5$ provided sufficient resolution of $\mathcal{X}_t$.

4. Implementation

4.1. The Model

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



(a) (b)

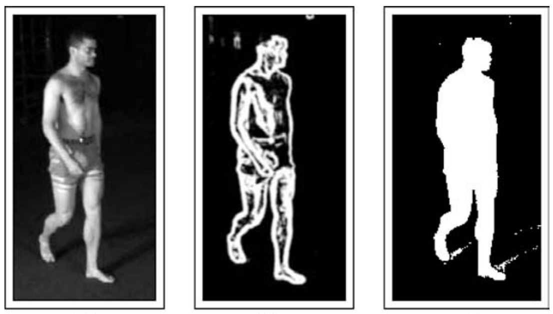
Figure 4. The model is based on a kinematic tree consisting of 17 segments (a). Six degrees of freedom are given to base translation and rotation. The shoulder and hip joints are treated as sockets with 3 degrees of freedom, the clavicle joints are given 2 degrees of freedom (they are not allowed to rotate about their own axis and are assumed to be coupled) 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 $\mathbf{X} = \{x_1 \ldots x_{29}\}$. The model is fleshed out by cones with elliptical cross-sections (b).

4.2. The Weighting Function

The basic annealed particle filter is a general optimisation tool and can be used for a variety of purposes (for another application see Deutscher et al. (2002)) with different weighting functions. In the present work we have constructed the weighting function on the basis of two image features—edges and foreground silhouette—chosen for their joint virtues of simplicity (i.e. easy and efficient to extract), and a degree of invariance to imaging conditions. While these features are not fully general (in particular the silhouette relies on a knowledge of the background which may not be available in more general environments) they suffice for our purposes. Even without a large degree of background clutter distracting edge measurements, there remains a challenging, multi-modal search problem because of self occlusions and foreground clutter (i.e. unmodelled markings on the target). Other features such as optic flow could equally be used.

4.2.1. Edges. 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 (Fig. 5(b)) in which each pixel is assigned a value related to its proximity to an edge.

P1: xxx
International Journal of Computer Vision KL3179-05/5384380 September 29, 2004 20:18



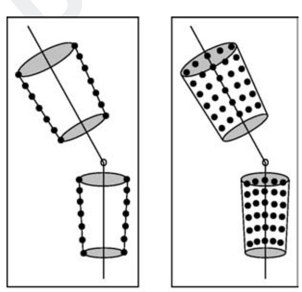
(a) (b) (c)

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 it proximity to an edge. The foreground is segmented using thresholded background subtraction to produce the pixel map (c) used in the weighting function.

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

$$\Sigma^e(\mathbf{X}, \mathbf{Z}) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - p_i^e(\mathbf{X}, \mathbf{Z})\right)^2 \qquad (14)$$

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 Fig. 6(a).



(a) (b)

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 truncated cones that have been projected into the image. The sampling points for the foreground segmentation measurements are taken from a grid within these occluding contours.

4.2.2. Silhouette. 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 Fig. 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 (Fig. 5(b)), and an SSD is computed

$$\Sigma^r(\mathbf{X}, \mathbf{Z}) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - p_i^r(\mathbf{X}, \mathbf{Z})\right)^2 \qquad (15)$$

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 truncated cones as seen in Fig. 6(b).

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

$$w(\mathbf{X}, \mathbf{Z}) = \exp -(\Sigma^e(\mathbf{X}, \mathbf{Z}) + \Sigma^r(\mathbf{X}, \mathbf{Z})). \quad (16)$$

An equal weighting to each component was determined empirically.

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

$$w(\mathbf{X}, \mathbf{Z}) = \exp -\left(\sum_{i=1}^{C}\left(\Sigma_i^e(\mathbf{X}, \mathbf{Z}) + \Sigma_i^r(\mathbf{X}, \mathbf{Z})\right)\right) \qquad (17)$$

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 Fig. 7.

5. Results

Two examples illustrate the system: in the first a subject walks in a circle as seen in Fig. 9; in the second the subject steps over a box, turns 180° on the spot before stepping over it again as seen in Fig. 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

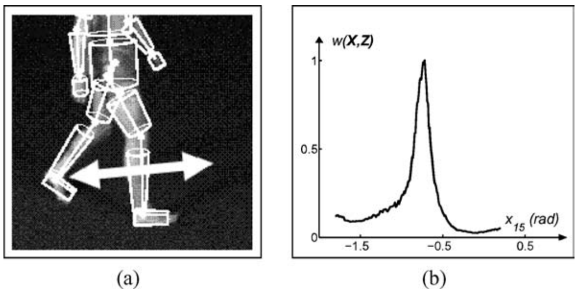


Figure 7. Example output of the weighting function obtained by varying only component x_{15} of $\mathbf{X}$ (the right knee angle) using the image and model configuration seen in (a). The function is highly peaked around the correct angle of -0.7 radians (b).

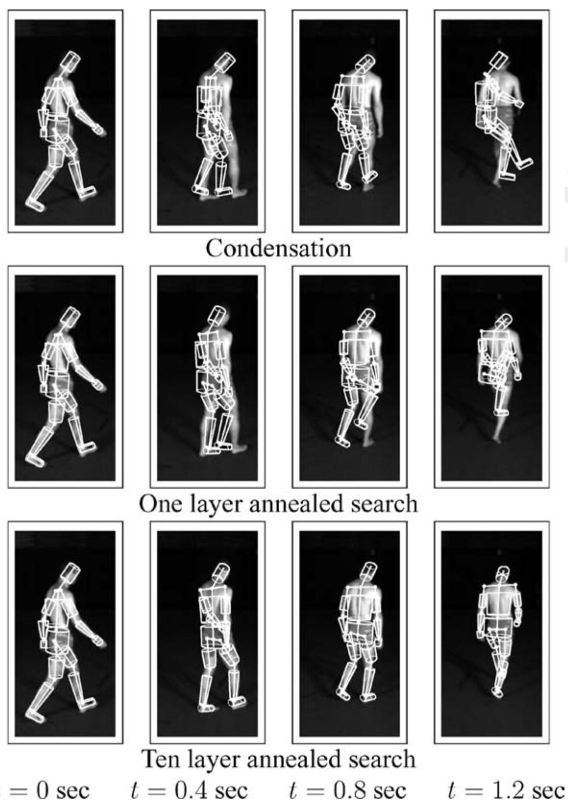


Figure 8. A comparison of condensation with the annealed particle filter. At top the results of tracking with 4000 particles using standard condensation can be seen. Tracking gradually deteriorates until terminal failure after 1.2 seconds. Experiments with 40000 particles were carried out taking over 30 hours to process just 4 seconds of video, still with negative results. An annealed search using 4000 particles with one layer fairs little better (middle), also suffering terminal failure after 1.2 seconds. An annealed search using 400 particles and 10 layers (i.e., 4000 weighting function evaluations per frame) tracks very well.

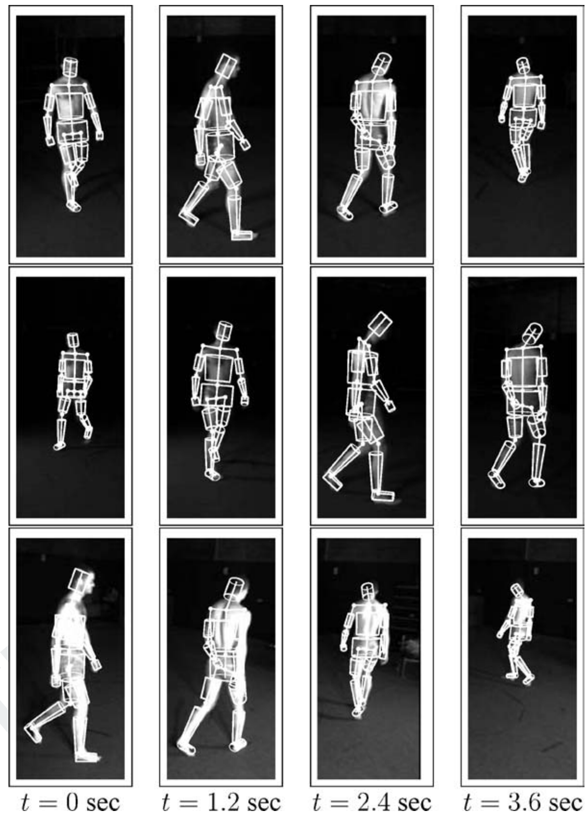


Figure 9. Walking in a circle. Using three cameras (arrayed here from top to bottom) a person is tracked over 4 seconds while walking in a circle. The tracker maintains an accurate lock throughout. 10 annealing layers were used with 200 particles for this sequence.

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 Fig. 8. Direct comparison is complicated by the fact that in Annealed Particle Filtering we use a simplified weighting function rather than a "correct" likelihood taking expected clutter into account (such as is derived in Blake and Isard (1998)). For this experiment the likelihood for Condensation comprised the edge based likelihood of Blake and Isard (1998), fused with a silhouette observation. The pose shown in each frame is the sample mean of the particle set. The one layer annealed search represents a similar experiment. It differs from Condensation in using the simplified weighting function (exactly the same as for the full Annealed Particle Filter experiment), and in propagating only the mode of the distribution between frames. The former difference accounts, remarkably, for a four-fold increase in speed

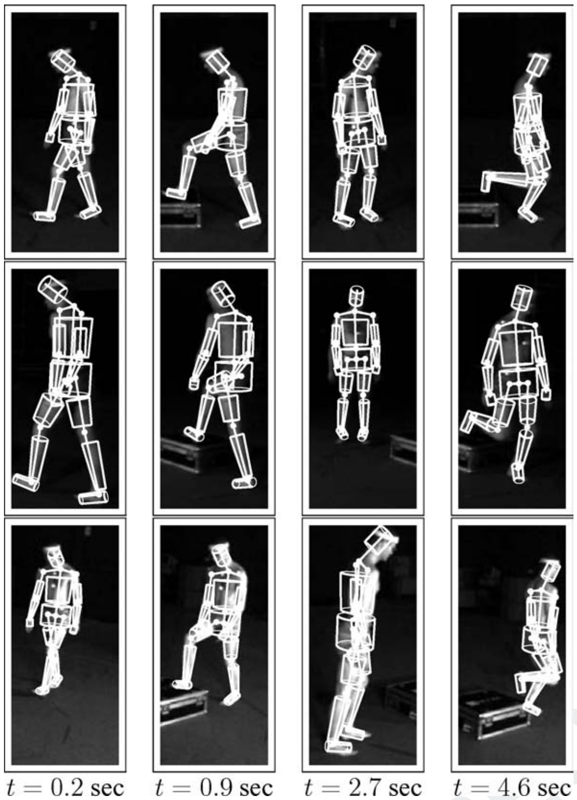


Figure 10. Stepping over a box. Using three cameras (arrayed here from top to bottom) a person is tracked over 5 seconds while stepping over a box, turning around and stepping over the box again. The tracker maintains an accurate lock throughout. 10 annealing layers were used with 200 particles for this sequence.

of execution. The final part of the experiment shows tracking using the full Annealed Particle Filter.

Each algorithm used a total of 4000 likelihood evaluations. In the final case this was divided as 400 particles over 10 layers. It was found in practice that good results on this sequence could be achieved with as few as 100 particles. While not being a strictly fair comparison between Condensation and the Annealed Particle Filter, the experiment gives an indication of the improved tracking performance of the APF given equivalent computational resources.

6. Algorithm Extensions: Hierarchical Search

The Annealed particle filter (APF), introduced in the previous sections, directly addresses the problem of searching high-dimensional configuration spaces, and has been demonstrated to be an effective and robust tracking framework for Human Motion Capture. However it remains a computationally intensive technique. The promise of further improvements is held out by the fact that the model is structured as a kinematic tree.

One way to reduce the effective volume of 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 restricting the rest of the search. This straightforward idea has been applied in a heuristic fashion by (among others) Gavrila and Davis (1996), who localised the torso using colour cues and used this information to constrain the search for the limbs, and more recently by Mikic et al., who first locate the head in order to limit their subsequent search. Although this approach is usually sound, without the assistance of colour cues (or other labelling cues) it is often very hard independently and reliably to localise specific body parts in realistic scenarios. Furthermore, failure of the first heuristic search can easily lead to catastrophic, unrecoverable failure.

A more formal approach to hierarchical search was proposed by MacCormick and Isard (2000). That work applied partitioned sampling to tracking articulated objects, but crucially assumed that the configuration space can be sliced so that one can construct an observation density for each dimension of the model configuration vector—effectively that it is possible independently to localise separate parts of an articulated model.

When using all but the simplest kinematic models, the optimal partitioning may not be obvious and it may indeed change over time as the degree of interaction between different segments of a model changes—such as when the legs cross during walking. Rather than impose a hierarchy on the search, we seek instead to develop a method for soft or fuzzy partitioning in which there is no need to commit to a particular hierarchy. Cham and Rehg (1999) capture this spirit in describing a search which is sequential in the degrees of freedom of the body. Their crucial innovation is to allow the order to be flexible: the search for body parts is conducted on a "best"-first basis, where best is defined as the component which can be found with minimum effort, usually meaning minimum variance.

While motivated by similar desires, our solution is rather different from theirs. Our approach improves upon and extend the APF in two ways. First we introduce a means to make the diffusion step in the APF adaptive, so that effort is not wasted in those places where the algorithm is already confident of doing well, and is concentrated on localising parts whose location is uncertain. The effect of this can be interpreted as a hierarchical search strategy which *automatically* partitions the search space in a soft way, without any explicit representation of the partitions (Section 6.1). Second, we introduce a crossover operator (similar to that found in Genetic Algorithms) which improves the ability of the tracker to search different partitions in parallel (Section 6.2).

We present results for simple examples to demonstrate the new algorithm's implementation and effectiveness, and show that these measures together increase the tracker efficiency by a factor of 4 and increase agility of the motion that can be tracked.

We apply the tracker to the complex problem of Human Motion Capture with 34 degrees of freedom. Extra degrees of freedom have been added to the model in Fig. 4 in the back (2) to allow arching that would not normally be encountered in everday walking (and was not neceeary in our ealier experiments), in the neck (1) to account for head nodding, and the clavicles are given independent motion (2 each).

6.1. Adaptive Diffusion and Hierarchical Partitioning

Consider the simple task of tracking an articulated arm as seen in Fig. 11. The arm consists of four segments, each joined by a swivelling joint with one end rooted on the spot. A configuration of the arm is described by an instance of the state variable $\mathbf{x} = (x_1, x_2, x_3, x_4)$. The weighting function $w(\mathbf{z}, \mathbf{x})$ required for the APF is computed by a Sum of Squared Differences (SSD) measure between a model template and a silhouette image (the detail to the regional correlation portion of the observation model in Eq. (15)).

The set $\mathcal{S}_{t,m}$ is initialised with particles uniformly distributed over a range of $\mathbf{x}$ that we know to contain the actual position of the arm. This results in a large and similar variance for each parameter of $\mathbf{x}$ over all the particles in $\mathcal{S}_{t,m}$ as can be seen in Fig. 12(a).

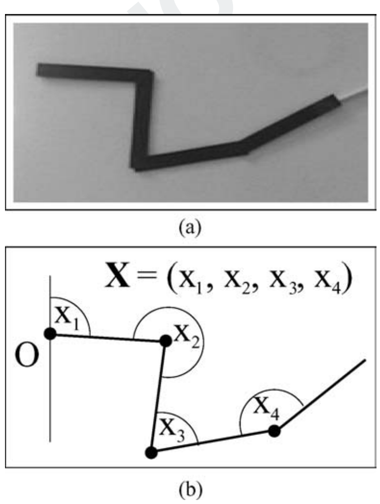


Figure 11. A planar articulated arm with 4 DOF is shown (a). It consists of four links connected by swivelling joints and rooted at **O**. The configuration of the arm is described by $\mathbf{x} = (x_1, x_2, x_3, x_4)$ as seen in (b).

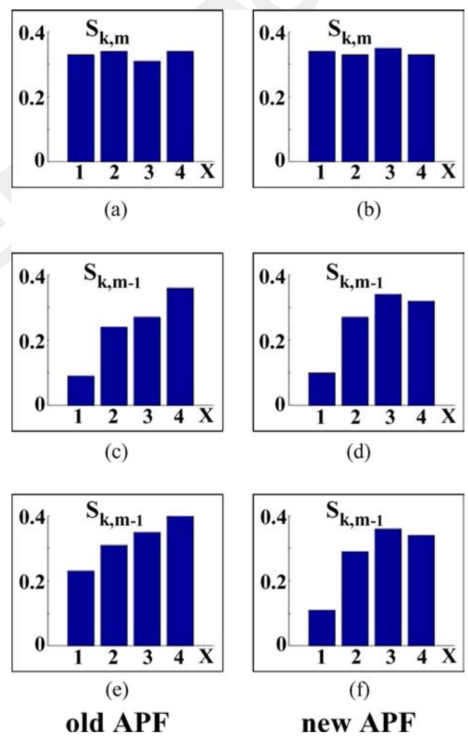


Figure 12. Parameter variance over one annealing layer: new APF vs. old APF. On the left graphs a, b and c plot the variance of each parameter of $\mathbf{x} = (x_1, x_2, x_3, x_4)$ through the first annealing run of the APF when tracking the articulated arm seen in Fig. 11. Graphs d, e and f show the same information for the improved APF as described in Section 6.1. Graphs a and d show the variances of the initial set $\mathcal{S}_{t,m}$, displaying equal variances for each parameter. Graphs b and e show the variances of the set $\mathcal{S}_{t,m-1}$ before the addition of diffusion noise. Note that in both b and e, x_1 has a very small variance indicating advanced localisation, however the variance of x_2, x_3 and x_4 has been reduced only a little. Up to this point the algorithms are the same and any differences between b and e are random. After the addition of noise in the original APF the localisation of x_1 has been greatly degraded as seen in graph c, however when noise is added in proportion to each parameter's variance the localisation of x_1 is preserved as seen in graph f.

After calculating a weight $\pi_{t,m}^{(i)}$ for each particle using $w_m(\mathbf{z}_t, \mathbf{s}_{t,m}^{(i)})$ we then proceed to Step 4 of the APF and draw N particles from $\mathcal{S}_{t,m}^{\pi}$ with replacement and probability proportional to each particle's weight.

Consider the set $\mathcal{S}_{t,m}$ so produced before the addition of any noise. In a typical annealing run the individual parameters of each particle were found to have variance as detailed in Fig. 12(b). Note here that the variance of x_1 has been greatly reduced while the other parameters x_2, x_3 and x_4 have been hardly reduced at all. The variance of any parameter can be considered (with a number of acceptable caveats) to be directly related to the degree to which the optimal value for that parameter has been localised. Figure 12(b) shows that x_1 has been localised down to a very small area of its range simply because it dominates the topology of the search space whereas each particle's values for x_2, x_3 and x_4 had very little influence on whether it was selected or not. In effect we see here an automatic partitioning of the state space into soft partitions according each parameter's topological dominance.

The weakness of the original APF (indeed any particle filter) arises with the addition of diffusion noise to each particle upon selection. According to Eqs. (7) and (13) an equal amount of noise should be added to each parameter. This results in a parameter variance profile like that seen in Fig. 12(c) with the localisation of x_1 seen in Fig. 12(b) all but wiped out by the excessive addition of noise.

If instead the amount of randomness added to the parameters of each selected particle is proportional to the variance of that parameter over the entire set of particles, these gains will be protected from disruption. Instead we will arrive at the situation seen in Fig. 12(f) where enough noise has been added to each parameter to allow the thorough diffusion of the particles into the spaces between repeatedly selected particles, but not enough to increase the variance of any given parameter which would erase any localisation gains made up to that point.

If this new method for determining the elements of $\mathbf{P}_i$ (the covariance matrix of $\mathbf{B}$ from Eq. (7)), is continued through all the annealing layers we can see that each parameter is localised in turn, with some degree of overlap as seen in Fig. 13. This can be compared to the pattern of variance reduction for the original APF algorithm seen in Fig. 14. This is exactly the kind of hierarchical soft partitioning that was desired and no explicit partition boundaries or functions were required.

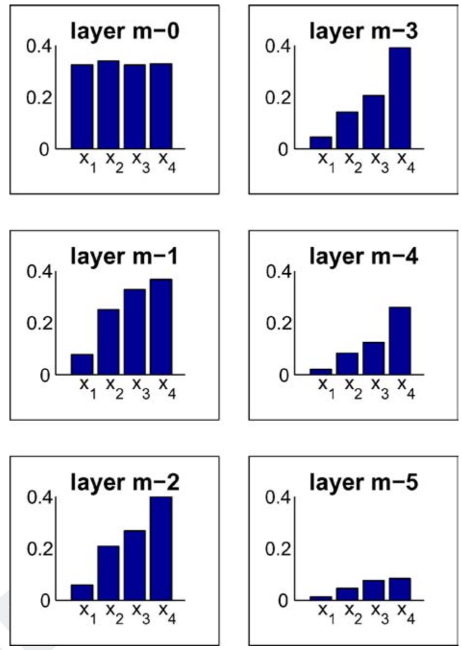


Figure 13. Variance reduction with the improved APF. Here we see the orderly reduction of each of the four parameter's variances from most dominant (x_1) to least dominant (x_4) over 6 layers of the annealing process while tracking the simple articulated arm. Using the improved APF results in a 2-fold increase in efficiency over the classical APF. Tracker efficiency was measured by the minimum number of particles needed to successfully track the articulated arm over 40 frames.

Sminchisescu and Triggs (2001) independently arrived at a very similar idea, although in that work they were concerned with most effective use of particles between frames in order to recover from "ambiguous" poses.

The changes to the APF are almost trivial, and can be formalised as follows. Step 4 of the APF algorithm described in Section 3 is amended so that at layer m, $\mathbf{P}_m$ is set to be proportional to the covariance of the particles in $\mathcal{S}_{t,m}$ as it exists before the addition of noise, i.e..

$$\mathrm{P}_m \propto \frac{1}{N}\sum_{i=1}^{N}\left(s_{t,m}^{(i)} - s_{t,m}^{av}\right)\cdot\left(s_{t,m}^{(i)} - s_{t,m}^{av}\right)^T. \quad (18)$$

where $\mathbf{s}_{t,m}^{av}$ is the sample mean of the particle set.

Using this modification enabled successful tracking with the APF with fewer than half the number of particles; i.e. a 2-fold increase in efficiency.

P1: xxx
International Journal of Computer Vision KL3179-05/5384380 September 29, 2004 20:18

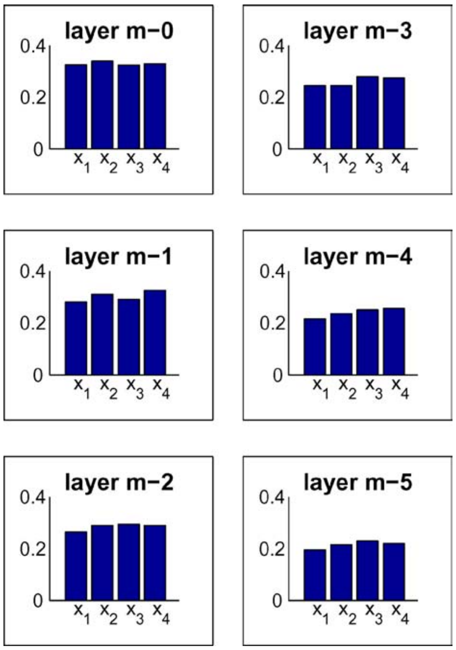


Figure 14. Variance reduction with the conventional APF. The even reduction in variance over 6 layers of the annealing process is shown in contrast to Fig. 13. There is little evidence of hierarchical partitioning and more annealing layers will be required to find the optimal configuration.

6.2. *A Crossover Operator and Parallel Partitions*

Now consider the articulated object found in Fig. 15 which consists of two articulated arms joined at a stationary hinge. This configuration is a much simplified version of that found in Human Motion Capture when using a model with arms and legs.

The soft hierarchical partitioning described in Section 6.1 provides some increase in efficiency over conventional APF when applied to tracking this assembly, localising x_1 and x_4 together, then x_2 and x_5 and finally x_3 and x_6. However if we were to decouple the search space and localise each arm independently the computational effort required for tracking would be reduced considerably.

One possibility, of course, would be to introduce a hard partition between the two arms and conduct two separate searches. However, in keeping with our philosophy of adaptive partitioning, we seek to avoid commitment to specific partitions.

Many people comment on the similarity between particle filters and Genetic Algorithms. Both employ a set (population) of particles (individuals) coded by a state vector (genetic sequence) from which the best particles (individuals) are chosen to be propagated to the next time-step (generation) in the hope of finding the maximum of some function (fittest possible individual).

One glaring difference between GA's and a typical particle filter is the lack of a crossover operator in the particle filter which in a conventional GA is meant to simulate the breeding of individuals and the sharing of genetic information. The use of the crossover operator encourages the survival of short, highly fit sections of the parameter space known in some GA literature as building blocks. This is done in the hope that when highly fit building blocks are brought together they will have a good chance of forming a very fit complete individual. These building blocks are effectively optimised in parallel without any specification of their boundaries or appropriate building block (partition)

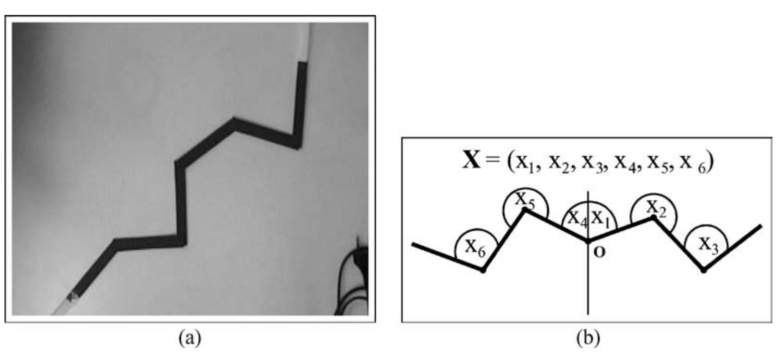


Figure 15. A pair of planar articulated arms consisting of 3 segments each and each rooted to point **O** (as seen in b) are used to demonstrate the effectiveness of the crossover operator. The configuration of the arms is described by $\mathbf{x} = (x_1, \ldots, x_6)$ as seen in (b).

P1: xxx
International Journal of Computer Vision KL3179-05/5384380 September 29, 2004 20:18

weighting functions, exactly the kind of behaviour we are looking for.

We now describe how to incorporate the crossover operator into the framework of the APF and examine the effect via a simple example.

6.2.1. Inclusion of the Crossover Operator in the APF. The inclusion of the crossover operator can be formalised as follows. In Step 4 of the APF (as described in Section 3) at annealing layer m, the ith particle of $\mathcal{S}_{t,m-1}$ is created by drawing two particles from $\mathcal{S}^{\pi}_{t,m}$ with probability proportional to their respective weights. Two parameter indices γ and ϵ are chosen randomly and the two selected particles $\mathbf{s}^{(a)}_{t,m} = (x^a_1 \ldots x^a_L)$ and $\mathbf{s}^{(b)}_{t,m} = (x^b_1 \ldots x^b_L)$ are combined to form the new particle $\mathbf{s}^{(i)}_{t,m-1}$ where

$$\mathbf{s}^{(i)}_{t,m-1} = \left(x^a_1, \ldots, x^a_\gamma, x^b_{\gamma+1}, \ldots x^b_\epsilon, x^a_{\epsilon+1}, \ldots, x^a_L\right). \tag{19}$$

Noise is then added to each particle as detailed in Section 6.1.

6.2.2. Testing the Crossover Operator. To assess the benefit to the crossover operator two articulated objects were tracked: the first (Fig. 11), was used in the experiment from Section 6.1, an un-branched articulated arm; the second as seen in Fig. 15 is two articulated arms rooted to the same position.

As seen in Fig. 16, the object consisting of branched arms was more effectively localised by the APF that employed the crossover operator whereas there was no difference when it was applied to the non-branched object. A good graphical illustration of what the crossover operator is actually doing—i.e. partitioning sections of the search space which can be tracked in parallel—is evident in Figs. 17 and 18 where the parameters localised best first are those closest to the root of the tree.

A good indication of the increased speed provided by the crossover operator when tracking branched objects is again the number of particles needed for successful tracking. This number was reduced by a factor of 2 with the introduction of the crossover operator.

6.3. Results for Full-Body Tracking

Although less clear-cut than the results for the "toy" example in the previous section, Figs. 19 and 20

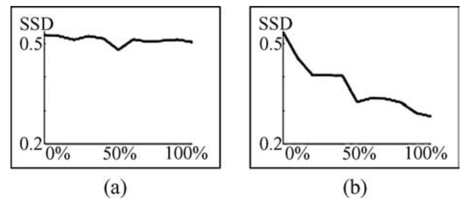


Figure 16. The crossover operator in action. The Sum of Squared Differences (SSD) match between model and image obtained after a set number of annealing layers is plotted against the percentage of particles generated using the crossover operator at each annealing layer. Graph (a) shows the result for the articulated arm seen in Fig. 11 where no benefit to using the crossover operator is seen although importantly no degradation in performance is seen either (i.e. the SSD does not increase). Graph (b) shows the result for the articulated arms seen in Fig. 15 where a steady improvement in tracking performance is seen when increasing the percentage of particles produced using the crossover operator. This shows that the crossover operator is able to decouple sections of the search space effectively and enables the APF to search them in parallel, improving tracker performance.

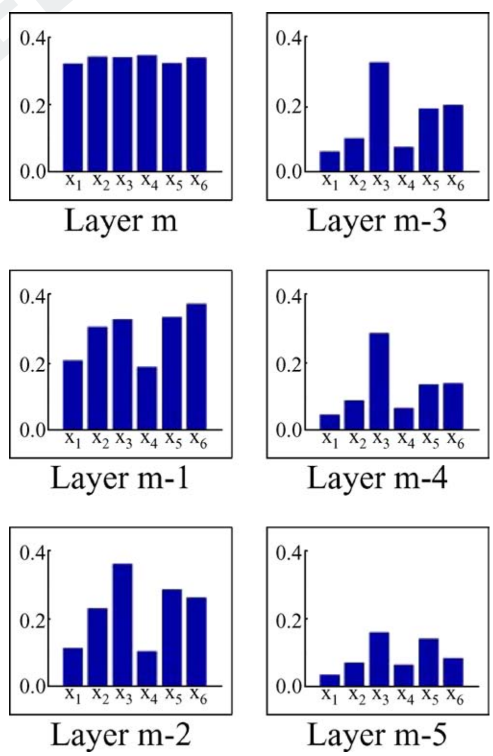


Figure 17. Variance reduction for the parallel arms. When the APF with crossover operator is applied to the articulated arms seen in Fig. 15 we get the pattern of variance reduction seen above. The graphs show the parameters describing each arm (x_1, x_2, x_3 and x_4, x_5, x_6) being localised in order of decreasing topological dominance, from the fixed point of the articulated arms, progressing outward.

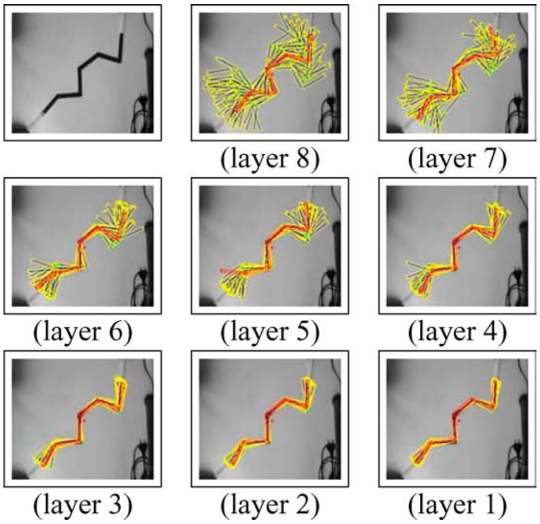


Figure 18. Particle distribution for the branched articulated arm over 8 annealing layers. The entire set of particles is drawn at each annealing layer. The hierarchical localisation of each model segment from the hinge joint outwards is clearly seen.

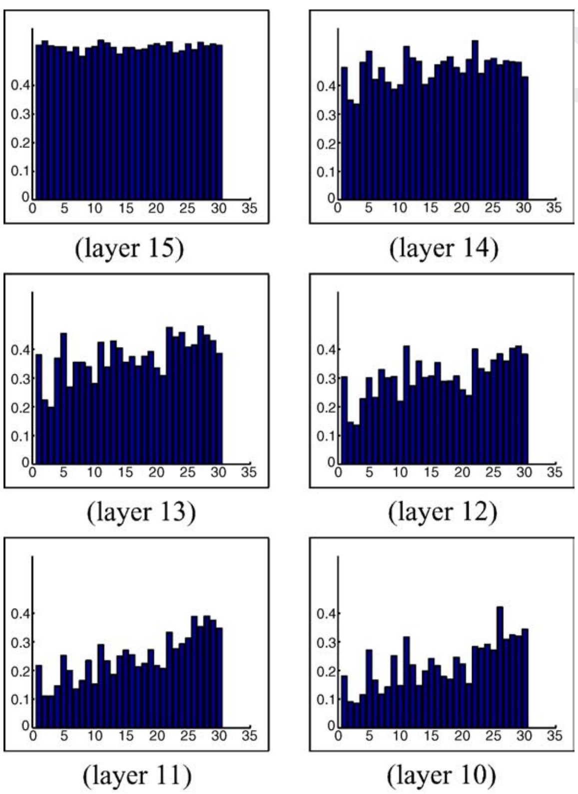


Figure 19. Annealed particle filter particle variance for a fully-body model. The difference in rates of variance reduction for each parameter can clearly seen. As expected a more complicated pattern of reduction than that seen for the simple articulated arm is evident.

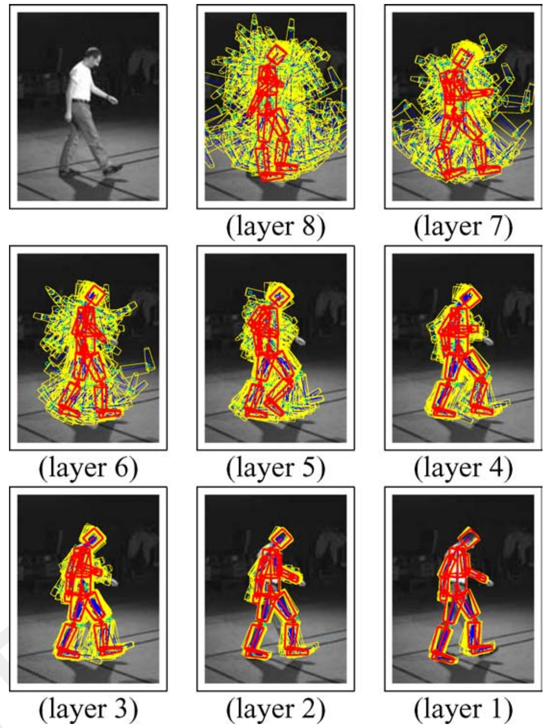


Figure 20. Particle distribution for a full-body model. The entire set of particles is drawn at each annealing layer for one frame. The hierarchical reduction of each parameter from torso rotation and translation out to the limb joint angles is evident.

show a similar process of variance reduction when the PAPF with crossover is applied to full-body tracking.

The algorithm was applied to a variety of challenging sequences of human movement including walking with turns (Fig. 21), running around in a random fashion (Fig. 22) and handstands (Fig. 23). The sequences for these experiments were generated using three evenly spaced cameras, calibrated and hardware synchronised.

We define successful tracking qualitatively as occurring when the algorithm locks onto the body and limbs for the duration of the sequence, returning sensible values (i.e. ones that can be used for re-animation, for example) for the pose and articulation parameters. Our tests measured the number of particles needed to achieve such successful tracking. This number represents a sensible measure of algorithm speed since the number of likelihood evaluations dominates the processing time.

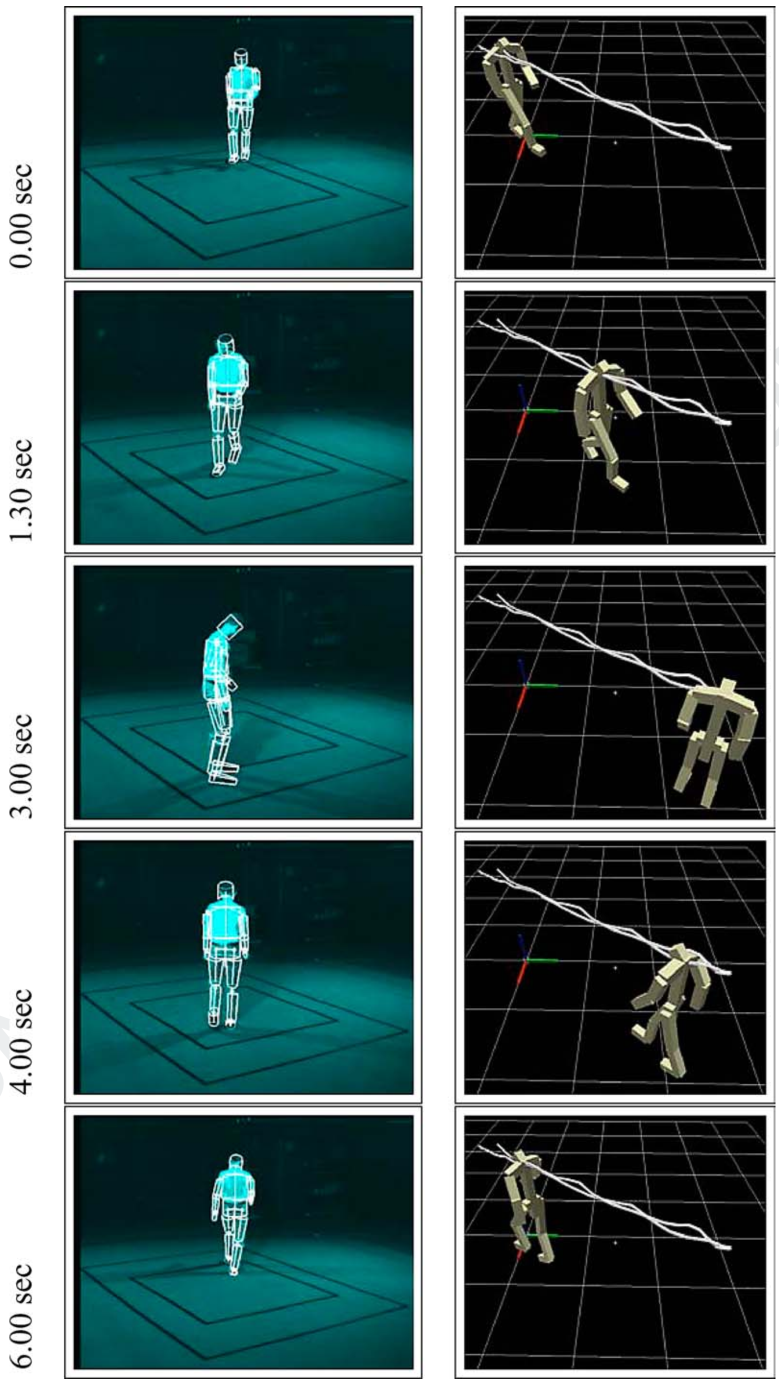


Figure 21. Tracking a walking person.

We observed an improved by a factor of 4 when comparing the new PAPF to the original APF (i.e. successful tracking achieved with one quarter the number of particles). As a result the PAPF required on average 15 seconds to process one frame whereas the APF required around 60 seconds when run on a single processor 1 GHz pIII Linux box.

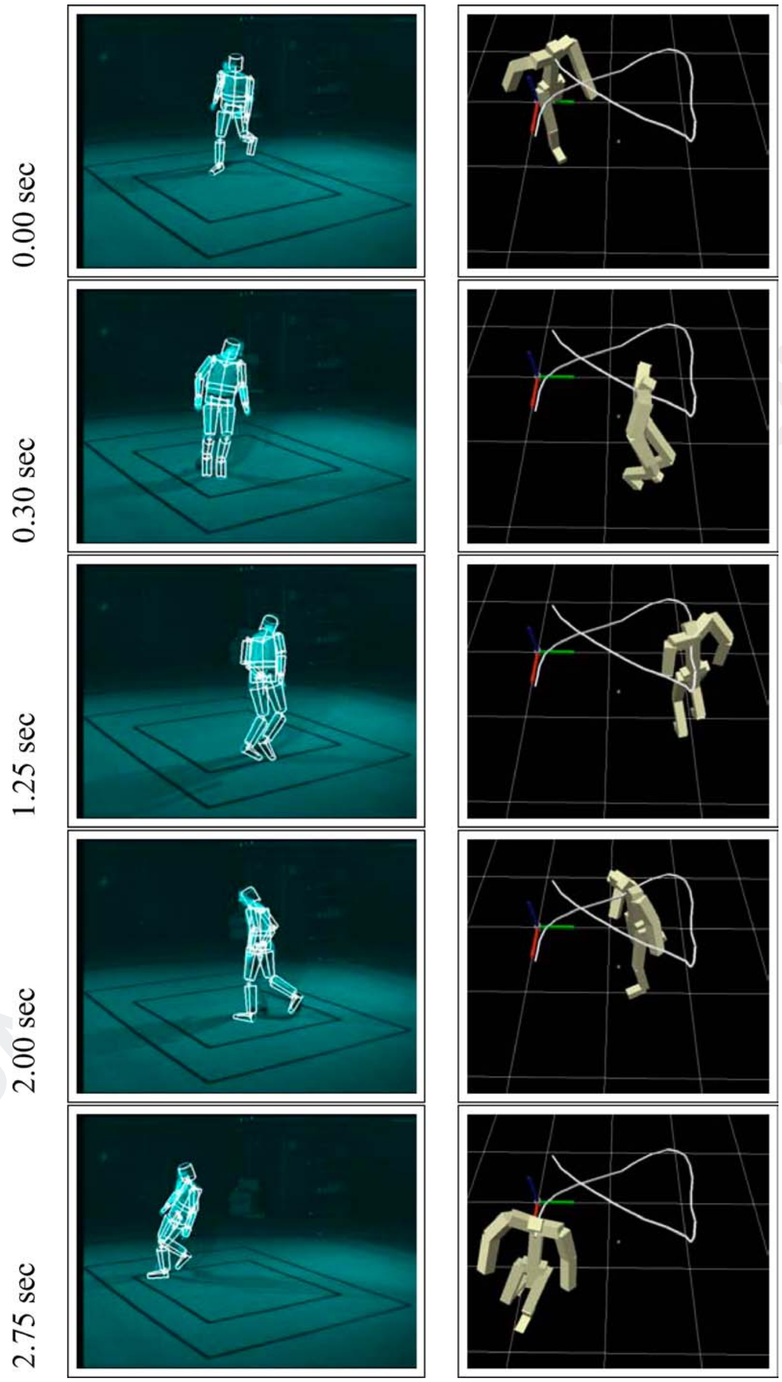


Figure 22. Tracking a running person.

We have also built a parallel implementation, in which particles are farmed out to independent processors which compute the weight/likelihood function. This achieves the sort of speed-ups that are to be expected, with processing time decreasing linearly in the number of processors (with a constant of proportionality around 0.8).

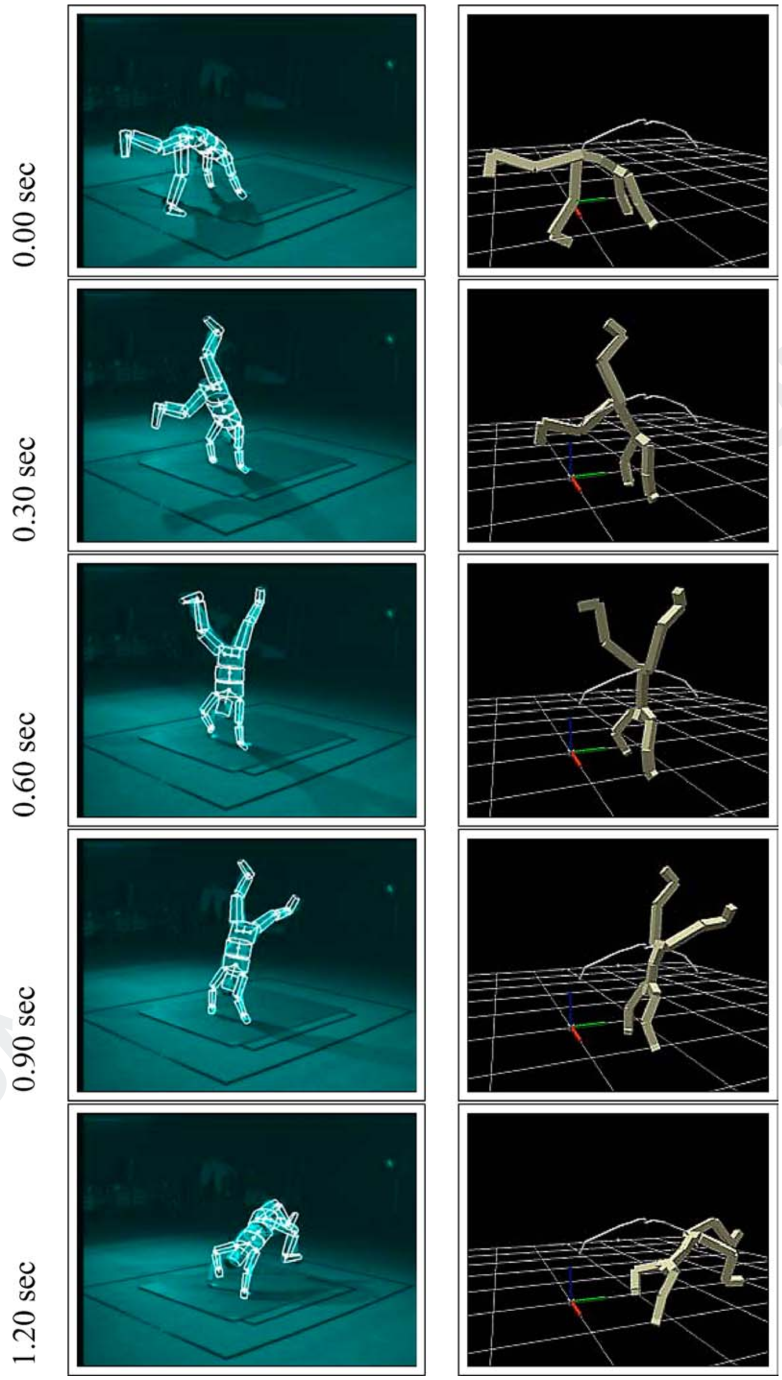


Figure 23. Tracking a person performing a hand-stand.

7. Discussion and Conclusion

We have developed a general algorithm for searching large configuration spaces which is more efficient than traditional particle filters, but which retains a number of their significant advantages. The algorithm has been applied to the problem of visually tracking a person in multiple cameras. In this context we have demonstrated reliable tracking of complex human motion using simple image features, and without the need for a strong dynamic model of the motion.

We have also introduced two novel improvements to the algorithm, soft hierarchical partitioning, and a crossover operator, which have the combined effect of improving performance and increasing efficiency.

The results, especially Figs. 21–23, show a robustness of tracking human motion achieved by very few other algorithms. Of particular note in the sequences shown are the points where the subject turns rapidly on the spot (shown in both Figs. 21 and 22), and the unusual and rapid motion of a handstand.

Our primary effort has been concentrated on the search technique. It seems clear that improvements in the modelling process, such as published in Plänkers and Fua (2003), and in dynamic modelling Sidenbladh et al. (2000), would improve tracking reliability and applicability further.

Though in the experiments shown the background lacks a large degree of clutter (but is not entirely clean either), tracking agile motions, even with multiple cameras, remains a difficult problem. We have performed experiments with other sequences with a greater amount of clutter with similar results, but the exact degree of clutter that can be tolerated is an open question. No doubt the use of background subtraction to obtain silhouette information assists in this significantly. The algorithm exhibits some robustness to errors in this data, but in cases where poor contrast results in poor silhouettes and a lack of edges we have observed tracker failure.

Our results to date have made use of 3 cameras, and tracking using a single camera raises issues with regard to ambiguity. Experiments with using the APF monocularly (Lyons, 2002) suggest that in the monocular case further sophistication in the placement of particles is required to overcome the inherent ambiguities and avoid all associated local minima. Some progress in this respect has been made recently by Sminchisescu and Triggs (2002, 2003).

Acknowledgments

This work was supported by Vicon Systems Ltd. and EPSRC grant GR/M15262. We would also like to thank Andrew Blake, Ben North, Andrew Davison and David Murray, for many useful discussions and the anonymous referees for insightful comments.

Note

1. Note, for example, that although (Blake and Isard, 1998) derives the full multi-modal likelihood model for edge-normal observations in the presence of clutter, the implementation makes a much simplified assumption of a unimodal likelihood for each individual observation.

References

Blake, A. and Isard, M. 1998. *Active Contours*. Springer.

Cham, T.-J. and Rehg, J.M. 1999. Dynamic feature ordering for efficient registration. In *Proc. 7th Int'l Conf. on Computer Vision*, Corfu, vol. 2, pp. 1084–1091.

Cham, T.-J. and Rehg, J. M. 1999. A multiple hypothesis approach to figure tracking. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 239–245.

Deutscher, J., Blake, A., North, B., and Bascle, B. 1999. Tracking through singularities and discontinuities by random sampling. In *Proc. 7th Int. Conf. on Computer Vision*, vol. 2, pp. 1144–1149.

Deutscher, J., Blake, A., and Reid, I.D. 2000. Articulated body motion capture by annealed particle filtering. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 126–133.

Deutscher, J., Davison, A.J., and Reid, I.D. 2001. Automatic partitioning of high dimensional search spaces associated with articulated body motion capture. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 669–676.

Deutscher, J., Isard, M., and MacCormick, J. 2002. Automatic camera calibration from a single manhattan image. In *Proc. 7th European Conf. on Computer Vision*, Copenhagen, vol. 4, pp. 175–188.

Drummond, T. and Cipolla, R. 2001. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *Proc. 8th Int'l Conf. on Computer Vision*, Vancouver, pp. 315–320.

Gavrila, D. and Davis, L.S. 1996. 3d model-based tracking of humans in action: A multi-view approach. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 73–80.

Harris, C.G. 1992. Tracking with rigid models. In *Active Vision*. A. Blake and A. Yuille (Eds.), MIT Press: Cambridge, MA.

Isard, M.A. and Blake, A. 1996. Visual tracking by stochastic propagation of conditional density. In *Proc. 4th European Conf. on Computer Vision*, Cambridge, England, p. 343–356.

Kirkpatrick, S., Gellatt, C.D., and Vecchi, M.P. 1983. Optimisation by simulated annealing. *Science*, 220(4598):671–680.

Lyons, D. 2002. A qualitative approach to computer sign language recognition. Master's thesis, University of Oxford.

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

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

MacCormick, J. and Isard, M. 2000. Partitioned sampling, articulated objects and interface-quality hand tracking. In *Proc. 6th European Conf. on Computer Vision*, Dublin, vol. 2, pp. 3–19.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. 1953. Equations of state calculations by fast computing machine. *J. Chem. Phys.*, 21:1087–1091.

Mikic, I., Trivedi, M., Hunter, E., and Cosman, P. 2001. Atriculated body posture estimation from multi-camera voxel data. In *Proc.*

of the IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 455–462.

Neal, R.M. 2001. Annealed importance sampling. *Statistics and Computing*, (11):125–139.

Plänkers, R. and Fua, P. 2003. Articulated soft objects for multi-view shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10).

Au: Pls. provide page range

Sidenbladh, H., Black, M.J., and Fleet, D.J. 2000. Stochastic tracking of 3D human figures using 2D image motion. In *Proc. 6th European Conf. on Computer Vision*, Dublin, vol. 2, pp. 702–718.

Sidenbladh, H., Black, M.J., and Sigal, L. 2002. Implicit probabilistic models of human motion for synthesis and tracking. In *Proc. 7th European Conf. on Computer Vision*, Copenhagen, vol. 1, pp. 784–800.

Sminchisescu, C. and Triggs, B. 2001. Covariance scaled sampling for monocular 3d body tracking. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 447–454.

Sminchisescu, C. and Triggs, B. 2002. Hyperdynamics importance sampling. In *Proc. 7th European Conf. on Computer Vision*, Copenhagen, vol. 1, pp. 769–783.

Sminchisescu, C. and Triggs, B. 2003. Kinematic jump processes for monocular 3d human tracking. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 69–76.

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

Wachter, S. and Nagel, H. 1999. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192.