

Parametric annealing: a stochastic search method for human pose tracking

Prabhu Kaliamoorthi
Nanyang Technological University
Singapore
prab0009@e.ntu.edu.sg

Ramakrishna Kakarala
Nanyang Technological University
Singapore
ramakrishna@ntu.edu.sg

Abstract

Model based methods to marker-free motion capture have a very high computational overhead that make them unattractive. In this paper we describe a method that improves on existing global optimization techniques to tracking articulated objects. Our method improves on the state-of-the-art Annealed Particle Filter (APF) by reusing samples across annealing layers and by using an adaptive parametric density for diffusion. We compare the proposed method with APF on a scalable problem and study how the two methods scale with the dimensionality, multi-modality and the range of search. Then we perform sensitivity analysis on the parameters of our algorithm and show that it tolerates a wide range of parameter settings. We also show results on tracking human pose from the widely-used Human Eva I dataset. Our results show that the proposed method reduces the tracking error despite using less than 50% of the computational resources as APF. The tracked output also shows a significant qualitative improvement over APF as demonstrated through image and video results.

1. Introduction

Tracking an articulated object like a human body with many degrees of freedom is an active research area in the vision community. A large body of research on human pose tracking and estimation suitable to different application areas is found in the literature [12]. The model-based generative method to tracking humans [14, 5, 6, 16, 3, 13, 17, 8] is a critical method with applications in animation, sports and medical motion analysis. Relying on a kinematic model to support the tracked hypothesis it remains one of the most accurate ways to track a human. However the accuracy comes with a significant computational overhead.

Model based generative methods model a human by a kinematic tree controlled by a fixed set of parameters. The imaging process is modeled by a projection operation and pose estimation is formulated as a state estimation problem that aims to minimize the disparity between the actual ob-

servation and the generated image as a function of the parameters of the kinematic model. Every evaluation of the disparity function incurs a significant computational overhead which can be considered to be of the order of the resolution and the number of cameras used for tracking.

Existing literature on the method either approach the problem as that of a Bayesian filtering problem [4] or as an optimization problem. Since the observation is not a random vector correlated with the hidden state, as is the case in filtering problems, Bayesian methods either resort to local optimization [3, 16] to recover and track the modes or choose a nonparametric method like particle filter [14]. In practice, however, both these methods need a significant number of evaluations of the disparity function, resulting in a high computational overhead to track.

Alternate approaches [8, 13, 7] formulate the problem as a search for the global mode of the likelihood assuming the disparity to be the negative log likelihood. This line of techniques originate from the Annealed Particle Filter (APF) [7]. APF is a generic procedure applicable to several types of input data including silhouettes from video streams and 3D reconstructions of human. Moreover it does not rely on learned prior dynamics [15], hence it remains an attractive algorithm and is largely considered to be the state-of-the-art in model-based human pose tracking.

APF reveals that annealing can be a powerful tool when applied to high dimensional multi-modal problems. However, we observe that the full potential of annealing is not realized by the APF. Since the cost of each likelihood evaluation is very high, there is a need to extract as much information as possible from each sample. Our procedure, which we refer to as parametric annealing (PA), is aimed at reducing the total number of samples by reusing samples across annealing layers and by using an adaptive parametric density to generate new samples. Our experiments show that it is capable of tracking more accurately with less than 50% of the number of samples required by APF. In this paper, we study various properties of our method including how it scales with the number of modes and dimensions, and also examine its sensitivity to parameter settings. A preliminary

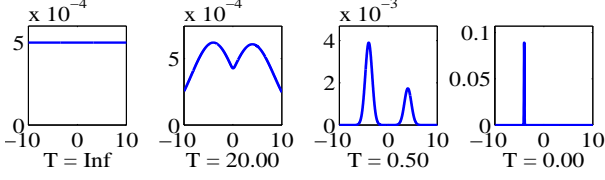


Figure 1: The plots shows the effect of annealing on a Mixture of Gaussians as the temperature T is varied from ∞ to 0. Annealing temperature is displayed below the plot.

version of our paper appears in [10].

The rest of the paper is organized as follows. Section 2 provides a background introduction to annealed particle filters. Section 3 highlights the problems with APF and motivates the need for our method. Section 4 provides a detailed description of our method. Section 5 compares the proposed method with APF on a simple scalable problem and performs sensitivity analysis on the parameters of our method. Following that, we provide a summary of the tracking results using data from Human Eva I dataset. Section 6 indicates the future work and concludes the paper.

2. Background

In this section we review the background of annealed particle filters to motivate the need for our method and to enable the exposition of it.

Particle filter [9] is a well-known technique in computer vision for tracking. It propagates the uncertainty about the tracked object non parametrically using a set of weighted samples. Let x_t and z_t be random variables corresponding to the state of the object and observation at time t . Let ψ be the set of N samples x_i and their corresponding normalized weights π_i that represent the distribution of x_t . The notation $z_{t:1}$ is used to indicate the set of observations till time t . Formally

$$p(x_t | z_{t:1}; \psi) = \sum_{i=1}^N \pi_i \delta(x_t - x_i) \quad (1)$$

Applying the first order Markov and the sensor assumptions [9] one can obtain the posterior at time $t+1$ by multiplying the likelihood with the prior, where the prior is obtained by convolving posterior from time t with the motion prior. Assuming the motion prior $p(x_{t+1} | x_t)$ to be a combination of a function $f(x_t)$ and Gaussian uncertainty of covariance Σ , results in a prior density shown below

$$p(x_{t+1} | z_{t:1}; \psi) = \sum_{i=1}^N \pi_i \mathcal{N}(f(x_i); \Sigma) \quad (2)$$

Particle filters use the prior density as a proposal density for importance sampling[11] from the likelihood. Multiplying the likelihood by the prior provides the posterior at

time $t+1$. This results in a four step procedure for tracking i.e. re-sample, drift, diffuse and evaluate[9]. The first three steps can be shown to generate samples from the prior density (2), and the last step can be shown to perform importance sampling and the necessary multiplication to obtain the posterior. The algorithm hinges on the critical assumption that the prior density has a good overlap with the likelihood for the importance sampling to be effective.

Annealing has the effect of transforming any distribution from a uniform distribution to a delta function in the global maximum as the temperature is changed from ∞ to 0. Figure 1 shows the effect of annealing on a multimodal distribution. Formally, the process of annealing can be expressed as below

$$f(x, T) \propto p(x)^{\frac{1}{T}} \quad (3)$$

where $p(x) : \mathbb{R}^d \mapsto \mathbb{R}$, is a non-negative function with a finite integral and T is the annealing temperature.

APF[7] incorporates annealing into particle filtering by exploiting the fact that a distribution at a higher temperature effectively has a wider support (as can be observed in Figure 1), and hence could be used as a proposal distribution for importance sampling from the same distribution at a lower temperature. Hence in a single iteration (that corresponds to a time interval t), it performs several particle filtering steps referred to as layers. It assumes $f(x_t)$ to be x_t in successive layers (since the distribution is not translated by annealing) and the variance to be gradually decreasing by a fraction α^m (where m is the annealing layer and $\alpha < 1$) to account for the narrow and peaked objective that results from annealing (see Figure 1). Annealing is performed by a simple schedule by controlling the number of unique resampled particles to be approximately half of that which exist in the previous layer. Consequently, at the end of a fixed number of layers, the set of samples that represented a multimodal density in the start represent a delta distribution in the end. Hence the expected state of the set of samples approximates the global maximum with high accuracy. The high level steps done in APF are as follows.

1. N samples $x_i, i \in \{1, \dots, N\}$ are obtained by sampling from $\mathcal{N}(\hat{x}_t; \Sigma)$, where \hat{x}_t is the estimate of the state in last iteration.
2. The likelihood y_i is evaluated at sample x_i .
3. A suitable annealing temperature T_m that ensures only 50% samples will survive resampling is estimated.
4. Normalized sample weights are obtained. $\pi_i = y_i^{\beta_m} / \sum_{i=1}^N y_i^{\beta_m}$, where $\beta_m = 1/T_m$
5. N new samples are generated by sampling (resample and diffuse) from $\sum_{i=1}^N \pi_i \mathcal{N}(x_i; \alpha^m \Sigma)$.
6. Repeat from step 2 for a fixed number of layers.

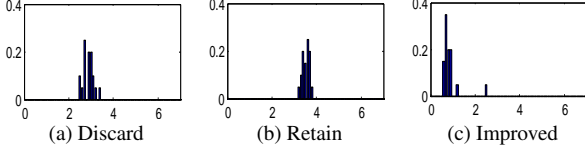


Figure 2: Test demonstrating effects of reusing samples. The subfigures a, b and c correspond to the configuration that discards samples, the one that retains them and the one that retains along with a deep annealing schedule. It can be observed that reusing makes the search less effective (b), which is overcome by augmenting it with a deep annealing schedule (c).

7. The global maximum is estimated as the expected state of the weighted samples.

The critical aspect of APF is that it is capable of tracking with significantly fewer number of samples. Studies done on human pose tracking show that APF is capable of tracking with 1000 [15] samples in comparison to 10,000 samples that are required by particle filters [14].

3. Motivation

The APF algorithm generates N new samples in each layer. However, samples from previous layers are discarded. Since the evaluation of likelihood incurs a significant overhead, one would want to retain the relevant samples from previous layers. The most natural way to do this would be to simply retain the samples x_i along with the corresponding likelihoods y_i , and raise them to a new annealing temperature T_m for the layer before normalization. However, this does not work well in practice.

To demonstrate the effect of simply reusing, we conducted an experiment. We considered the log likelihood to be a quadratic, $-\frac{1}{2}x^T x$, where $x \in \mathbb{R}^{30}$. To verify how well APF searches through the state space we started from a fixed state and searched for the optimum. We compared two configurations of APF: the original configuration that discards samples, and a second configuration which retains them by raising the likelihood to the appropriate β_m for the layer before normalization. To measure how effectively the methods locate the optimum, we obtained the distribution of the L2 norm of the state estimated by APF. The histograms of the L2 norm are shown in Figure 2a & 2b. It can be observed that when samples are discarded, the state estimate is closer to the true optimum.

The test demonstrates the problems with retaining samples. The critical insight, however, is that the problems with reusing samples are more than overcome by choosing a “deep” annealing schedule that consistently reduces the annealing temperature, rather than a simple schedule which ensures 50% samples survive resampling. The precise formulation of a deep annealing schedule as a power

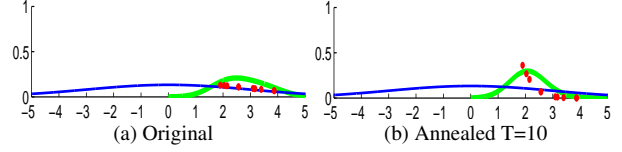


Figure 3: Example demonstrating the effect of annealing(color). It can be seen how annealing shifts the probability mass towards regions that improve the objective.

law, which is a novel aspect of our method, is given in Section 4. The improvement is shown in the histogram in Figure 2c which augments retaining samples with our proposed deep annealing schedule. The result can be best explained by analyzing another simple example. Figure 3a shows a 1D Gaussian density in blue. A set of weighted samples were obtained by diffusing samples around $x = 3$. The samples are shown in red and the kernel density approximated by the samples using a Gaussian kernel is shown in green. If the samples were to be re-sampled and diffused with the same Gaussian kernel, it is equivalent to generating samples from the density shown in green. Figure 3b shows the effect of annealing on the kernel density. It can be observed that annealing effectively shifts the kernel density mass towards regions that improve the search objective. Consequently, when enabled with a suitable annealing schedule, new samples are generated in areas of the state-space that improve the objective. Expressed differently, a good annealing schedule ensures that the worthless samples which end up in areas of the state-space with poor objective don’t adversely impact the search process by quickly driving their normalized weights to zero in successive layers.

In addition, one would expect to have a slow annealing schedule when tracking articulated objects since their likelihoods are multimodal and fast annealing is known to converge to local optima. APF can be interpreted as generating new samples from a kernel density with a Gaussian kernel of gradually reducing noise covariance. It is well-known that the kernel density approximated by a set of samples is highly sensitive to the type of kernel. Therefore such a fixed diffusion schedule is very sensitive to the parameters, more so when the annealing is slow and longer. This is evident from the two-fold reduction [6] in the number of samples needed to track when using an adaptive kernel. We take the idea of adaptive diffusion density a step further, and use an intermediate parametric form which is inferred from the set of samples to act as a substitute for the kernel density i.e. instead of resample and diffuse with a kernel, new samples are generated from a parametric model. The motivation is that, with an intermediate parametric form, the density would adapt even better.

Inferring an intermediate parametric form has an added advantage when the set of samples in the last layer of annealing still represent a multimodal density. Such a scenario

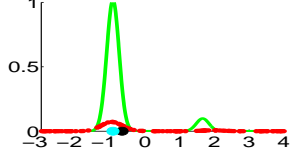


Figure 4: Example demonstrating the effect of a parametric form on the state estimate. The samples are shown in red, and an inferred parametric form is shown in green. The expected state of all samples, and the maximum of the parametric form, are shown in black and cyan respectively. It can be seen how the expected state is a worse estimate of the true optimum at -1.

can be observed in Figure 4, it shows a set of samples in the last layer of annealing in red. A parametric density inferred from the set of samples is shown in green. The parametric density shows that the samples represent a multimodal density. It can be observed that the expected state of such a sample set deviates from the global maximum, and a better estimate is obtained from the parametric form. These considerations motivate the need for an improved procedure, which we refer to as parametric annealing. We describe our method in detail in the next section.

4. Parametric Annealing

Let $P(x)$ ($x \in \mathbb{R}^d$) be a general multi-modal distribution, and let ψ be the set of N weighted samples (x_i, π_i) that represent $P(x)$. Formally,

$$P(x) \approx p(x; \psi) = \sum_{i=1}^N \pi_i \delta(x - x_i) \quad (4)$$

Let $q(x; \theta)$ be a Mixture of Gaussians (MoG), parametrized by θ that approximates $p(x; \psi)$. We estimate the parameters of $q(x; \theta)$ using Expectation Maximization (EM) adapted to include the sample weights π_i . Formally,

$$q(x; \theta) \approx p(x; \psi) \approx P(x) \quad (5)$$

With this terminology, the high level steps done in each iteration of our algorithm are now described. Subsequently, the steps are compared to the APF, and the differences are highlighted.

1. N^0 samples $x_i, i \in \{1, \dots, N^0\}$ are obtained by sampling from $\mathcal{N}(\hat{x}_t; \Sigma)$, where \hat{x}_t is the estimate of the state in last iteration.
2. The likelihood y_i is evaluated for new samples x_i .
3. A suitable annealing temperature T_m is estimated using a deep annealing schedule.

4. Normalized sample weights $\pi_i^m = y_i^{\beta_m} / \sum_{i=1}^{N^m} y_i^{\beta_m}$, where $\beta_m = 1/T_m$ are estimated.
5. A parametric MoG approximation $q(x; \theta^m)$ is inferred from the weighted samples ψ using one EM iteration.
6. C new samples x_i are generated from the parametric model and combined with old samples.
7. Repeat from step 2 for M layers to simulate a very slow annealing.
8. The global maximum is estimated from the parametric model.

The main novelty of our procedure over APF is twofold.

- We don't discard samples in step 6, and moreover, we use a deep annealing schedule to enable effective search in step 3.
- We use a parametric model in steps 5 & 6 instead of the kernel diffusion density, and obtain the estimate from the parametric form in step 8.

Below, we provide details of the method, which may be skipped by those readers who want to consider the results in the next section.

The high level steps above include some of the parameters of the algorithm. The superscript m in any parameter indicates that it is a function of the layer, $m \in (1, \dots, M)$, where M is the number of annealing layers. The parameter N^m represents the total number of samples in layer m . As opposed to APF where the number of samples in each layer remain fixed, the number of samples in our method grows in successive layers since we don't discard samples. As we introduce C new samples in a layer and we start with N^0 samples, the number of samples in each layer N^m is $N^0 + mC$, bringing the total number of samples for an iteration to $N^0 + MC$.

We fix the number of mixture components in the MoG $q(x; \theta)$ to C , which is also the number of new samples introduced in a layer. Consequently, the parameters θ^m is the set of C means (μ_c^m) , covariances (Σ_c^m) and the mixture weights (ϕ_c^m) . The subscript c and the superscript m indicates that these parameters are dependent on the specific mixture component c and the layer m . The "E & M" steps [1] in the EM algorithm are well known. A regularizer ξ_m / ϕ_c^m is included in the M step update for Σ_c^m to ensure that the covariance matrices are full rank. Since we are also annealing the samples while inferring the parametric form we find that a gradually reducing regularizer is more effective than a fixed regularizer.

The annealing schedule is a critical part of our algorithm. We use a measure similar to particle survival rate α defined in [7] to control the annealing schedule. The parameter T_m

which is the annealing temperature for the layer m can be estimated from a predefined sequence of α_m by any local optimization technique. The parameters α_m and ξ_m (regularizer) define the annealing schedule. We define these parameters by the following power law

$$\xi_m = \alpha_m \xi \xi' \alpha_m, \quad \alpha_m = \eta \lambda^m \quad (6)$$

Where ξ is a $d \times d$ diagonal matrix with each diagonal element equaling half the maximum change in state estimate along that dimension. The significant difference between our schedule and that used by APF is that APF sets the parameter α_m to 0.5 for all m . Thus $\eta \in (0, 1)$, $\lambda \in (0, 1)$, N^0 , M and C are the external parameters used by our method. These parameters are dependent upon the state space dimension d and the density $P(x)$.

5. Analysis and comparison

5.1. Scaling properties

In this section we analyze how our method scales with the dimensionality, multi-modality and the range of search by comparing it with APF on a simple scalable problem. The problem used for the comparison is to let the two stochastic procedure search for the optimum state in a MoG. Inspired by [2], we used a generative method to model the MoG used as the search objective. We assume the search objective θ (which is a MoG) to be a random draw from a distribution $P(\Theta)$. The distribution $P(\Theta)$ is sampled using a generative method i.e., the mixture weights are drawn from a Dirichlet distribution of uniform prior, the inverse covariances are drawn from a Wishart distribution with d dimensional identity as scale matrix and the mean vectors are drawn from a d dimensional Gaussian with a scaled identity matrix (by scale s) as covariance and zero mean.

The variable d , the number of mixture components k and the scale of the covariance s control the dimensionality, multimodality and search range of the objective. We made both the procedures start from the origin of the euclidean space \mathbb{R}^d and search for the optimum. We considered the difference in the log likelihood between the end state and the start state to be a random variable I (for improvement in the log likelihood) that acts as an indicator of how well the procedure performs. Since the procedures are stochastic in nature we ran the procedure several times on different samples from $P(\Theta)$ and considered the mean and standard deviation of I to be indicative of the performance.

Figure 5 plots the scaling properties of the proposed method in comparison to the APF. We compared our method to two configurations of APF. The configurations are APF 1000 ($M = 5, N = 200$) and APF 500 ($M = 5, N = 100$), where M and N are annealing layers and particles per layer respectively. Our method was configured with 438 samples ($438 = N^0 + M * C = 150 + 24 * 12$).

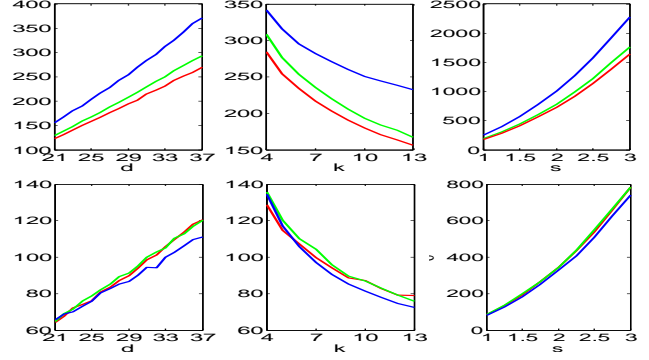


Figure 5: Scaling properties of the PA (blue) in comparison to the APF 1000 (green) and APF 500 (red). The plots shows the mean (top row) and deviation (bottom row) of improvement in log likelihood for different values of dimension d , number of components k and range parameter s respectively. It can be observed that our method improves the likelihood more than APF, and that the deviation in improvement is almost the same.

The mean and standard deviation of the random variable I is plotted against various values of dimension d , components k and scale factor s . Higher mean value of I and lower deviation are better. The result might seem to show that all procedures somehow perform better in higher dimensions and as the range increases, but this is not true. It is caused by the vast change in the log likelihood in higher dimensional Gaussian mixtures and over long search ranges. However, it can be observed that, in all three tests, our method improves the log likelihood better than APF despite using less than 50% of the samples. It either performs as good as or slightly better than APF with regards to the deviation of improvement.

5.2. Sensitivity analysis

Both our method, as well as APF, have quite a number of parameters. A standard set of parameters proposed in [7] is used in most studies involving APF [15]. However, how these parameters affect the tracking performance has not been studied. This motivated us to perform sensitivity analysis on the parameters of our own algorithm since it would provide insight into how those parameters affect tracking. We performed this study in the same objective as the previous experiment. The test was performed by varying each parameter η , λ , N^0 , M and C around a small range from the default parameter used in previous test and observing the mean and deviation of the variable I . Figure 6 shows the results of sensitivity analysis. It can be observed that our algorithm is largely insensitive to the initial number of samples N^0 . In fact, it slightly improves as the initial number of samples reduce, which may be understood by analyzing Figure 3. When the initial number of samples are higher, the

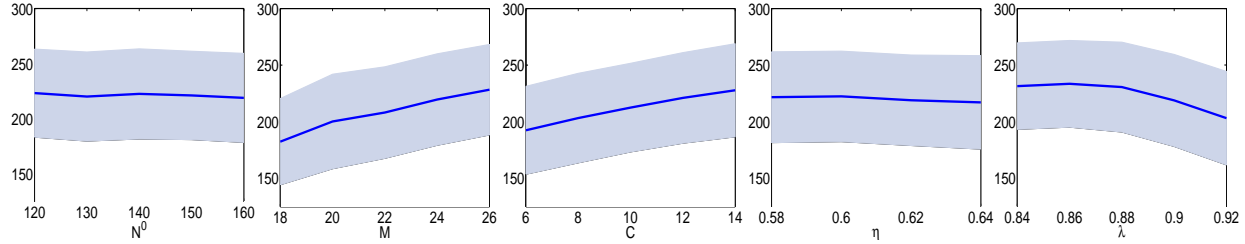


Figure 6: The plot shows the mean improvement and deviation for various settings of the parameters of our procedure. The parameters N^0 , M and C refer to initial number of samples, number of annealing layers and the number of new samples introduced per layer respectively. The parameters η and λ define the power law used to control the rate of annealing.

search is ineffective since the samples hold back the search procedure until they are annealed to a suitable temperature. Similar observations were made with human pose tracking, suggesting one might reduce the initial number of samples without losing performance.

An increase in the number of annealing layers M improves the performance, this justifies our choice of slow annealing. However it can be observed the deviation shows a slight decreasing trend as the number of layers reduce. We believe this is due to the cumulative effect of the randomness introduced in every layer. Hence a suitable trade-off between the two is necessary. An increase in the number of new samples C introduced in a layer improves the performance, this is expected since when there are more samples the search should be better. A decrease in the parameters η and λ that define the annealing schedule shows an improvement in performance, indicating that the deeper the annealing, the better the performance of the search procedure. This is expected since, as shown in Section 3, a deep annealing schedule is necessary to ensure that the new samples search the state space effectively.

5.3. Tracking Results

We compared the proposed method with two configurations of APF. a) 1000 particle configuration with 200 particles per layer and 5 layers b) 500 particle configuration with 100 particles per layer and 5 layers. Our method was configured with 438 samples ($N^0 = 150$, $M = 24$, $C = 12$). We used data from the Human Eva I dataset for the comparison. We review the overall results here which will be presented in detail elsewhere. Table 1 presents the results of the comparison from 6 different video sequences. Each tracking method was executed 10 times on a sequence and the time and ensemble average of the tracking error and the deviation are shown in the Table. We observe that, for all videos except one, our method has the lowest tracking error. Furthermore, the difference is significant for jogging sequences. It is notable that our method reduce despite the fact that it uses half the number of sample, and requires half the runtime as APF 1000.

Figure 7 shows the tracked model from all three meth-

ods superimposed on images used for tracking. We used the Subject 1 jogging sequence for the figure since, as evident from Table 1 it is the sequence that by a large extent discriminates the three configurations. It can be observed that the tracked output from our method is visibly closer to the Subject.

6. Conclusion & Future Work

In this paper we described a procedure that improves on APF for tracking a human from a video sequence. Using synthetic examples we demonstrate the critical problems with APF and show how they are overcome by the novel aspects of our procedure, which include reusing samples with a suitably deep annealing schedule, and also by inferring a parametric form from the samples. We then compared the proposed method to APF in a simple scalable problem and show that our method consistently performs better than APF. This was followed by sensitivity analysis showing that our algorithms' performance is largely insensitive to the parameter settings. Finally, we present human pose tracking results using data from the Human Eva I dataset that show the benefit of using our algorithm. We plan to explore techniques to optimally estimate the parameters for tracking and to understand how various attributes like the frame rate and the articulated motion affect the tracking performance.

References

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006. 4
- [2] D. M. Blei and M. I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1:121–144, 2005. 5
- [3] T.-J. Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2239–2245, 1999. 1
- [4] Z. H. E. Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics*, pages 1–69, 2003. 1
- [5] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Computer Vision*

Method	Number of Samples	Time averaged mean \pm standard deviation of the tracking error (mm)						Runtime per frame
		Subject 1		Subject 2		Subject 3		
		Walk	Jog	Walk	Jog	Walk	Jog	
APF	1000	119.2 \pm 25.6	179.8 \pm 18.8	106.3 \pm 16.7	122.6 \pm 7.4	132.7 \pm 3.9	115.4 \pm 10.2	32.49 s
APF	500	132.4 \pm 31.5	200.7 \pm 29.7	107.3 \pm 16.4	141.6 \pm 29.2	134.0 \pm 5.2	127.2 \pm 21.1	16.30 s
PA	438	93.8 \pm 12.4	142.4 \pm 14.5	110.2 \pm 15.3	114.4 \pm 8.9	132.3 \pm 4.4	109.8 \pm 12.2	14.18 s

Table 1: Overall statistics. Best results are shown in bold.

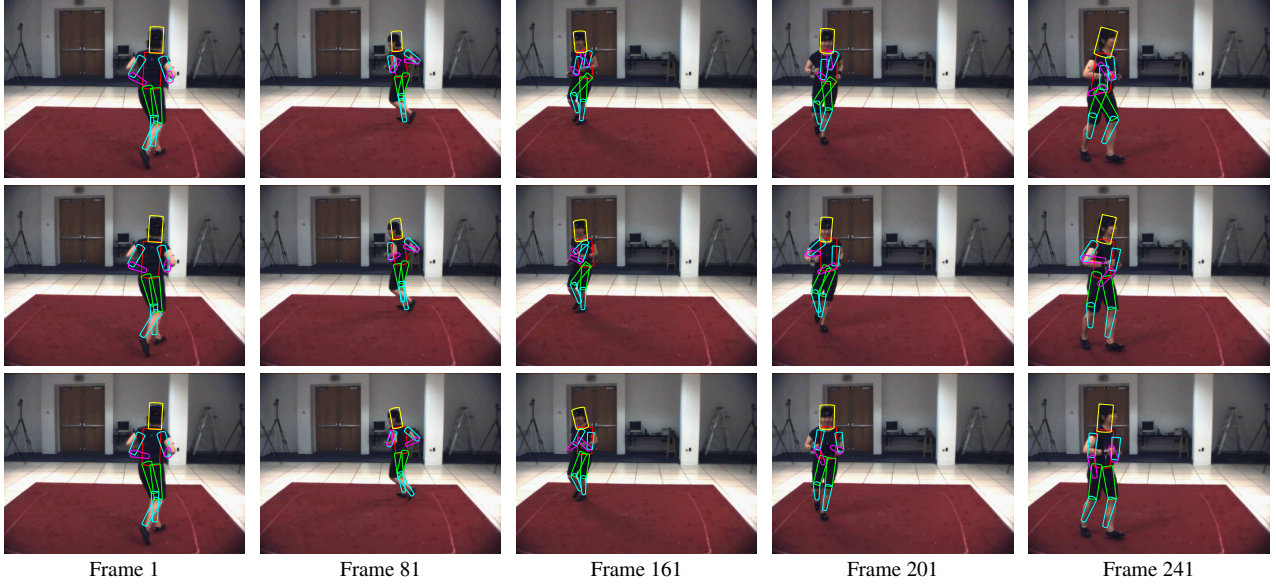


Figure 7: Qualitative comparison of tracked output on Subject 1 jogging sequence. First, second and third row correspond to APF 500, APF 1000 and PA 438 respectively. The model fit can be observed to be worst for APF 500, better for APF 1000, and best for PA 438.

- and Pattern Recognition (CVPR), volume 2, pages 126–133, 2000. [1](#)
- [6] J. Deutscher, A. Davison, and I. Reid. Automatic partitioning of high dimensional search spaces associated with articulated body motion capture. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 669–676, 2001. [1](#), [3](#)
- [7] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61:185–205, 2005. [1](#), [2](#), [4](#), [5](#)
- [8] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and filtering for human motion capture. *International Journal of Computer Vision*, 87:75–92, 2008. [1](#)
- [9] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998. [2](#)
- [10] P. Kaliamoorthi and R. Kakarala. Human pose tracking by parametric annealing. In *IEEE Computer Society conference on Computer Vision and Pattern Recognition, 2nd International Workshop on Human Activity Understanding from 3D Data (HAU3D)*, 2012. [2](#)
- [11] D. J. C. MacKay. Introduction to Monte Carlo methods. In *Proceedings of the NATO Advanced Study Institute on learning in graphical models*, pages 175–204, 1998. [2](#)
- [12] T. B. Moeslund, A. Hilton, and V. Krger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006. [1](#)
- [13] L. Raskin, E. Rivlin, and M. Rudzsky. Using gaussian process annealing particle filter for 3d human tracking. *EURASIP Journal on Advances in Signal Processing*, 2007. [1](#)
- [14] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision (ECCV)*, pages 702–718, 2000. [1](#), [3](#)
- [15] L. Sigal, A. O. Balan, and M. J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87:4–27, 2010. [1](#), [3](#), [5](#)
- [16] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22:371–392, 2003. [1](#)
- [17] X. Xu and B. Li. Learning motion correlation for tracking articulated human body with a Rao-Blackwellised particle fil-

ter. In *International Conference on Computer Vision (ICCV)*,
pages 1–8, 2007. [1](#)