IEEE Workshop on Motion and Video Computing, Orlando, Florida, Dec. 2002, pp. 151-158

Hybrid Monte Carlo Filtering: Edge-Based People Tracking

Eunice Poon[*†] **David J. Fleet**[*]

[*]Palo Alto Research Center, 3333 Coyote Hill Rd, Palo Alto, CA 94304
[†]Department of Electrical Engineering, Queen's University, Kingston, Canada K7L 3N6

Abstract

Statistical inefficiency often limits the effectiveness of particle filters for high-dimensional Bayesian tracking problems. To improve sampling efficiency on continuous domains, we propose the use of a particle filter with hybrid Monte Carlo (HMC), an MCMC method that follows posterior gradients toward high probability states, while ensuring a properly weighted approximation to the posterior. We use HMC filtering to infer the 3D shape and motion of people from natural, monocular image sequences. The approach currently uses an empirical, edge-based likelihood function, and a second-order dynamical model with soft bio-mechanical joint constraints.

1 Introduction

Statistical inefficiency often limits the effectiveness of Monte Carlo methods for probabilistic inference and Bayesian tracking. Most applications of particle filters, for example, have been limited to problems in which the number of state variables is relatively small. In high dimensions they quickly become computationally expensive as the required number of samples grows exponentially with dimension. One promising direction for improving statistical efficiency involves the use of Markov chain Monte Carlo (MCMC) updates after particle propagation [5, 23]. In experiments with synthetic data, it was shown that MCMC updates can produce Bayesian state estimates several orders of magnitude faster than conventional particle filters yet with similar estimator variance [5]. In this paper, we use hybrid Monte Carlo (HMC) to infer the 3D shape and motion of people from natural image sequences. The HMC filter uses an empirical edge-based likelihood function and a second-order dynamical model with soft bio-mechanical joint constraints.

2 Bayesian Filtering and People Tracking

The goal of Bayesian filtering is to compute the posterior probability distribution $\mathcal{P}_t \equiv p(\mathbf{s}_t \,|\, \mathbf{z}_{1:t})$ over a hidden state $\mathbf{s}_t$ at time t, conditioned on image observations, $\mathbf{z}_{1:t} \equiv (\mathbf{z}_1, ..., \mathbf{z}_t)$, up to time t. Like many tracking problems, we model the time-varying state as a Markov process, and we assume that observations are independent given $\mathbf{s}_t$. Then we may factor the posterior, $p(\mathbf{s}_t \,|\, \mathbf{z}_{1:t})$, to obtain

$$p(\mathbf{s}_t \,|\, \mathbf{z}_{1:t}) \;=\; \kappa\, p(\mathbf{z}_t \,|\, \mathbf{s}_t)\; p(\mathbf{s}_t \,|\, \mathbf{z}_{1:t-1})\,, \qquad (1)$$

where κ is a constant, independent of $\mathbf{s}_t$. Here, $p(\mathbf{z}_t \,|\, \mathbf{s}_t)$ is the likelihood function, and the prediction distribution, $p(\mathbf{s}_t \,|\, \mathbf{z}_{1:t-1})$, is easily shown to be

$$p(\mathbf{s}_t \,|\, \mathbf{z}_{1:t-1}) \;=\; \int p(\mathbf{s}_t \,|\, \mathbf{s}_{t-1})\; p(\mathbf{s}_{t-1} \,|\, \mathbf{z}_{1:t-1})\; d\mathbf{s}_{t-1}\;. \qquad (2)$$

When the posterior distribution is complex, nonGaussian, and multimodal, it is often appropriate to compute nonparametric approximations to it. Particle filters approximate the posterior using a discrete set of weighted states (or particles) [1, 7, 11, 12, 16]. Simple particle filters draw states, $\mathbf{s}_t^i$, directly from the prediction distribution in order to bound the search for high probability states. These states are then weighted so they properly approximate the posterior, rather than the prior from which they were drawn. The *importance weights*, w_t^i, are simply equal to the normalized likelihood values, $p(\mathbf{z}_t \,|\, \mathbf{s}_t)$, i.e., the probability that the current observations were generated by the hypothesized state. The resulting samples, $\mathcal{S}_t = \{\mathbf{s}_t^i, w_t^i\}_{i=1}^N$, are said to be properly weighted when sample averages approximate expectations under the posterior $\mathcal{P}_t$ [16]; i.e.,

$$\mathrm{E}_{\mathcal{S}_t}[f(\mathbf{s}_t)] \;\equiv\; \sum_{i=1}^{N} w_t^i f(\mathbf{s}_t^i) \;\stackrel{N\to\infty}{\longrightarrow}\; \mathrm{E}_{\mathcal{P}_t}[f(\mathbf{s}_t)]\,, \qquad (3)$$

for sufficiently smooth functions f.

The success of a Monte Carlo method depends on its ability to maintain a good approximation to the posterior. Following (3), one way to assess the quality of the filter is to examine the expected distance between the sample mean, $\mathrm{E}_{\mathcal{S}_t}[\mathbf{s}_t]$, and the true posterior mean, $\mu_t \equiv \mathrm{E}_{\mathcal{P}_t}[\mathbf{s}_t]$, over many runs of the filter; e.g.,

$$\mathrm{E}\left[(\mathrm{E}_{\mathcal{S}_t}[\mathbf{s}_t] - \mu_t)^2\right] = \frac{\alpha}{N}\,\mathrm{E}_{\mathcal{P}_t}\Big[(\mathbf{s}_t - \mu_t)^2\Big]\;. \qquad (4)$$

Here, α is often referred to as an *inefficiency factor*. If the N samples in $\mathcal{S}_t$ were drawn independently from $\mathcal{P}_t$ then it

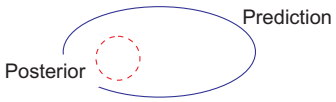

Figure 1. Particle filters draw independent samples from a prediction volume (solid ellipse) in order to find high probability states inside the posterior volume (dashed circle).

is straightforward to show that $\alpha = 1$. By comparison, in trying to find *posterior* states, particle filters draw independent samples from the *prediction* distribution. In this case, the effective number of independent samples will be given by the number of samples drawn which are also high probability states. As illustrated in Fig. 1, the expected fraction of high probability samples depends on the ratio of the effective volumes of the prediction density and the posterior density. Unfortunately, this ratio, and hence the required number of particles, grows exponentially with the dimension of the problem. A common measure of the *effective number of (independent) samples* is given by α/N and approximated as $N/\sum(w_t^i)^2$ [3, 15, 17].

One can reduce the number of particles by choosing a better prediction distribution, e.g., by improving the dynamical model or by finding a low-dimensional subspace in which the tracking can be performed [14, 22]. This is appropriate when low-dimensional representations are available. Other ways to obtain better proposals involve importance sampling, partitioned sampling [17], or sampling from low-level detectors in order to rapidly inject good hypotheses into the sample set [1, 13]. Deutscher et al. [6], Cham and Rehg [4], and Plankers and Fua [21] tackle the problem of tracking people in high dimensional spaces by following gradients to good hypotheses. Although such methods produce maximal-likelihood parameter estimates, they do not produce an approximation to the desired posterior. Even with multiple hypotheses [4], the samples are not likely to be properly weighted with respect to the posterior.

HMC is an MCMC method that follows the gradient of the posterior to good hypotheses, while designed to ensure that it draws fair samples from the posterior [8, 19]. When properly tuned, it allows for long trajectories through state space so the posterior can be sampled rapidly. Choo and Fleet [5] proposed an HMC filter that uses multiple Markov chains to explore multiple minima. They found the HMC filter to be several orders of magnitude more efficient than conventional particle filters. Sminchisescu and Triggs [23] proposed a variation on HMC that lowers the effective energy walls between neighboring extrema to facilitate the exploration of local maxima of the posterior. Although encouraging, both these papers did experimental work with synthetic data; Choo and Fleet required labeled moving light displays, while Sminchisescu and Triggs used hand-marked 2d joint locations. This paper, by comparison, describes progress toward an HMC filter for tracking 3D people directly from monocular, grayscale image sequences.

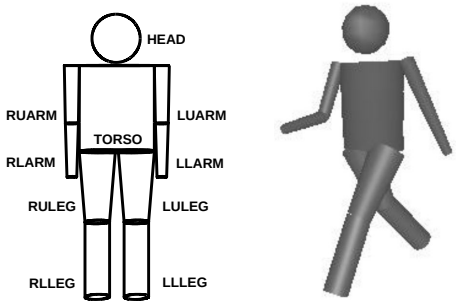

Figure 2. Human model: (left) frontal view. (right) 3D view.

3 Generative Model Formulation

Before discussing the HMC filter in detail, we begin by formulating our state-space model, along with the generative model for the observations and the temporal dynamics. These probabilistic formulations provide the foundation from which we derive the form of the likelihood function and the prediction distributions, as well as their gradients.

3.1 Articulated Human Model

Our human body model (Fig. 2) is an articulated collection of cylindrical parts. Each limb comprises two tapered cylinders with circular cross-sections. The torso is tapered with an elliptical cross section, and the head is spherical. Quadratic part definitions were chosen to simplify image projections of occluding boundaries.

Rigid transformations specify the relative positions and orientations of parts with respect to one another in a hierarchical manner. Each elbow and knee joint has one rotational degree of freedom. Each hip and shoulder joint has three degrees of freedom, represented using Euler angles. Finally, six degrees of freedom are used to specify the location and orientation of the torso with respect to a camera-centered coordinate system. The state, $\mathbf{s}_t$, therefore includes 4 variables for each limb, and 6 more for the torso.

3.2 Edge-Based Likelihood Function

The likelihood is derived from an empirical model of image structure in the neighborhood of occluding surface boundaries. Following [20], the observation density is determined from the response behavior of orientation-tuned, band-pass filters [10] that are steered to the orientation of the boundary. Nestares and Fleet [20] showed that, conditioned on the image position and orientation of a surface boundary, the responses of filters on the boundary, steered to the boundary orientation, were well modeled as functions of their complex-phase and amplitudes.

The density for a phase measurement, $\phi \in [0, \pi)$, conditioned on log amplitude ρ and the edge state, is well modeled by a mixture of a Gaussian and a uniform density [20]:

$$p(\phi \,|\, \rho, \mathbf{s}) \;=\; \epsilon(\rho)\; G(\phi; \mu, \sigma^2) \;+\; (1 - \epsilon(\rho))\; p_c \;, \qquad (5)$$

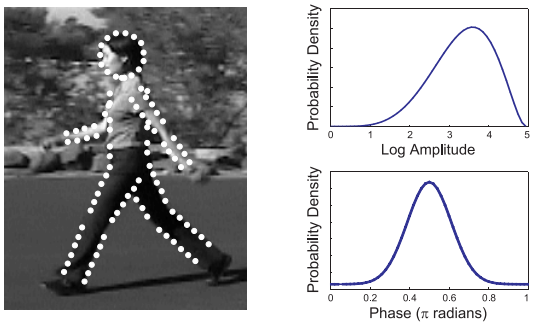

Figure 3. (left) Conditioned on the state, the white dots denote image locations where amplitude and phase responses are measured. (right) Examples of observation densities for log amplitude and phase measurements.

where $p_c = 1/\pi$ is the uniform outlier probability, the Gaussian mean and standard deviation can be fixed at $\mu = \pi/2$ and $\sigma = 0.15\pi$, and the mixing probability, $\epsilon(\rho)$, is well modeled as a linear function of ρ. The density for the normalized log amplitude, i.e., $\rho' = (\rho - \rho_{min})/(\rho_{max} - \rho_{min})$, is similarly well modeled by a Beta distribution [20]; i.e.,

$$p(\rho \,|\, \mathbf{s}) \;=\; \begin{cases} \kappa(a,b)\;(\rho')^{a-1}\,(1-\rho')^{b-1} & 0 < \rho' < 1 \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

where $\kappa(a,b)$ is the appropriate normalization constant. Fig 3 shows examples of the observation densities in (5) and (6).

Given the state $\mathbf{s}$ and the quadratic form of the body parts, it is relatively easy to solve analytically for the location and orientation of each part's occluding boundaries. Under pseudo-orthographic projection, the visible boundaries of each part lie in a plane that bisects the part and is perpendicular to the line of sight. Here, we first find the boundary end-points of each part given by by $\mathbf{p} = \mathbf{e} \pm r\,(\mathbf{e}\times\mathbf{c})/\|\mathbf{e}\times\mathbf{c}\|$, where $\mathbf{e}$ is an end-point of the part's cylindrical axis, r is the part radius, and $\mathbf{c}$ is the camera viewing direction. These endpoints are then projected into the image under perspective projection, yielding a polygonal approximation to the shape of each part. We then use the convexity of the parts to detect the regions of each part that occluded one another.

Along each part boundary we obtain measurements at equispaced pixel locations (see Fig. 3). For the visible edge segments, the likelihood function is simply the observation density for the band-pass filter responses in (5) and (6). An outlier noise process is used to model samples that are occluded from view. This yields a likelihood function, at image location i, for body part j, of the form:

$$L_{i,j} \;=\; \begin{cases} p(\phi_{i,j} \,|\, \rho_{i,j}, \mathbf{s})\; p(\rho_{i,j} \,|\, \mathbf{s}) & \text{if visible} \\ p_{occ} & \text{otherwise}\,, \end{cases} \qquad (7)$$

where p_{occ} is a constant occlusion probability, and $L_{i,j}$ is of course a function of the state $\mathbf{s}$.

Based on the observation model in (7), we formulate the edge-based log likelihood for the body in terms of the joint probability over measurements on all J body parts, normalized by the number of measurements n_j on each part:

$$\mathbf{L}(\mathbf{s}) \;=\; p(\{\phi_{i,j}, \rho_{i,j}\} \,|\, \mathbf{s}) \;=\; \prod_{j=1}^{J} \prod_{i=1}^{n_j} L_{i,j}^{1/n_j} \qquad (8)$$

Normalization with n_j and the constant occlusion probability in (7) are motivated by computational, rather than theoretical issues. They account for the lack of a background model for model comparison, and they broaden the likelihood somewhat so that narrow peaks are easier to find.

Unlike the particle filter, the HMC filter requires the gradient of the log likelihood. Given (8), the partial derivative of $\log L(\mathbf{s})$ with respect to the k^{th} state variable is

$$\frac{\partial \log \mathbf{L}(\mathbf{s})}{\partial s_k} \;=\; \sum_{j=1}^{J} \sum_{i=1}^{n_j} \frac{1}{n_j}\, \frac{\partial \log L_{i,j}}{\partial s_k}\,. \qquad (9)$$

The derivative for an individual measurement is given by

$$\frac{\partial \log L_{i,j}}{\partial s_k} \;=\; \frac{\partial \log L_{i,j}}{\partial \phi} \frac{\partial \phi}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial s_k} + \frac{\partial \log L_{i,j}}{\partial \rho} \frac{\partial \rho}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial s_k} \qquad (10)$$

if the sample is visible, and 0 otherwise. The phase derivative of $\log L_{i,j}$ is given by

$$\frac{\partial \log L_{i,j}}{\partial \phi} \;=\; \frac{-1}{p(\phi \,|\, \rho, \mathbf{s})} \left(\frac{\phi - \mu}{\sigma^2}\right) G(\phi; \mu, \sigma)\,,$$

and the derivative with respect to log amplitude is

$$\frac{\partial \log L_{i,j}}{\partial \rho} = \frac{\partial \epsilon(\rho)}{\partial \rho}\, \frac{G(\phi;\mu,\sigma) - 1}{p(\phi \,|\, \rho, \mathbf{s})} + \frac{a-1}{\rho - \rho_{min}} - \frac{b-1}{\rho_{max} - \rho}\,.$$

The phase and amplitude gradients with respect to spatial position in (10) are computed as in [9], and the derivative of measurement locations, $\mathbf{x}$, with respect to the state variable $\mathbf{s}_k$ is obtained by differentiating the projection of the limb boundaries onto the image plane.

3.3 Second-Order Stochastic Dynamics

To complete the generative model, the temporal dynamics specifies the stochastic evolution of states from one time to the next. Unlike previous approaches that assumed highly constrained models, such as walking [22], the goal here is a generic model of smooth 3D motion. Accordingly, we assume a simple second-order Markov process. Towards this end, it is convenient to define an augmented state $\mathbf{y}_t$ [2]:

$$\mathbf{y}_t \equiv \begin{bmatrix} \mathbf{y}_t^+ \\ \mathbf{y}_t^- \end{bmatrix} = \begin{bmatrix} \mathbf{s}_t \\ \mathbf{s}_{t-1} \end{bmatrix}. \qquad (11)$$

With this one can derive new filtering equations that are identical to those in (1) and (2) but with $\mathbf{s}_t$ replaced by $\mathbf{y}_t$:

$$p(\mathbf{y}_t \,|\, \mathbf{z}_{1:t}) \;=\; \kappa\; p(\mathbf{z}_t \,|\, \mathbf{y}_t{}^+)\; p(\mathbf{y}_t \,|\, \mathbf{z}_{1:t-1})\,. \qquad (12)$$

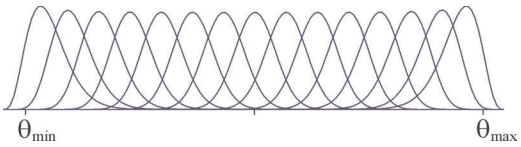

Figure 4. Examples of Beta noise densities with modes centered at the predicted joint angle (θ).

The likelihood depends solely on the current body configuration; the full augmented state is used only for prediction.

The dynamics consist of a deterministic prediction, $\hat{\mathbf{y}}_t^+ = \alpha_1 \mathbf{y}_{t-1}^+ - \alpha_2 \mathbf{y}_{t-1}^-$, and additive process noise, η:

$$\mathbf{y}_t^+ \;=\; \hat{\mathbf{y}}_t^+ + \eta(\hat{\mathbf{y}}_t^+)\,, \;\text{ and }\; \mathbf{y}_t^- \;=\; \mathbf{y}_{t-1}^+\,. \qquad (13)$$

Here, the coefficient α_2 controls the extent to which velocity influences the prediction; i.e., we can rewrite the prediction as $\hat{\mathbf{y}}_t^+ = (\alpha_1 - \alpha_2)\,\mathbf{y}_{t-1}^+ + \alpha_2\,\mathbf{v}_{t-1}$ where $\mathbf{v}_{t-1} = \mathbf{y}_{t-1}^+ - \mathbf{y}_{t-1}^-$ is a measure of the velocity from time $t-2$ to time $t-1$.

For torso position and rotation variables we use Gaussian and wrapped Gaussian process noise respectively. For the remaining joint angles the process noise must respect the appropriate bio-mechanical joint limits. As depicted in Fig. 4, we let the noise η have a Beta density, with fixed variance and a mode that is bounded softly above and below. In effect, we simply set the mode to be the prediction $\hat{\mathbf{y}}_t^+$, but clipped at the physical joint limits. As shown in Fig. 4, the densities become increasingly skewed as the clipped prediction state parameter approaches the joint boundaries.

3.4 Prediction Distribution

With Monte Carlo approximations to the posterior, the integration in (2) yields a mixture model. For N equally weighted particles $\{\mathbf{y}_{t-1}^i\}_{i=1}^N$, the prediction density is

$$p(\mathbf{y}_t \,|\, \mathbf{z}_{1:t-1}) \;=\; \frac{1}{N}\sum_{i=1}^{N} p(\mathbf{y}_t \,|\, \mathbf{y}_{t-1} = \mathbf{y}_{t-1}^i)\,. \qquad (14)$$

With the HMC filter we also require the gradient of the log prediction distribution:

$$\frac{\partial \log p(\mathbf{y}_t \,|\, \mathbf{z}_{1:t-1})}{\partial \mathbf{y}_{t_k}} \;=\; \frac{1}{N}\,\frac{\sum_{i=1}^{N} \frac{\partial p(\mathbf{y}_t \,|\, \mathbf{y}_{t-1} = \mathbf{y}_{t-1}^i)}{\partial \mathbf{y}_{t_k}}}{p(\mathbf{y}_t \,|\, \mathbf{z}_{1:t-1})}\,,$$

where the partial derivatives of the transition probability are found by differentiating the process-noise densities.

The HMC filter begins each frame with mixture model prediction. The particles used in our current HMC filter are the expected values from the individual HMC chains described below.

4 Hybrid Monte Carlo Filtering

Like particle filters, the HMC filter uses a Monte Carlo approximation to $\mathcal{P}_t \;\equiv\; p(\mathbf{s}_t \,|\, \mathbf{z}_{1:t})$. The sample states

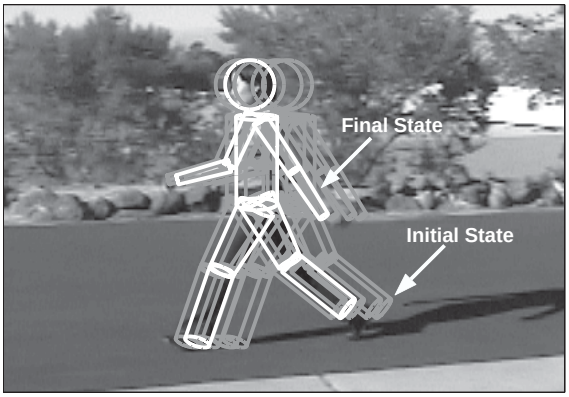

Figure 5. HMC simulation from the initial to final states, with all variables except the torso held fixed.

are, however, drawn using an MCMC procedure. A single Markov chain could eventually explore the entire state space, but it often requires many samples to move between different modes of the posterior. Therefore, following [5], we use several Markov chains at each time. Let there be M Markov chains, each with $R+1$ samples, where $\mathbf{s}_t^{c,i}$ denotes the i^{th} sample from the c^{th} Markov chain.

The first step is to find M initial states. Ideally one draws these states from an approximate posterior to reduce the number of *burn-in* samples (i.e., the time before chains reaches equilibrium and yield samples from $\mathcal{P}_t$). Here, we find initial states using a sampling-importance-resampling step. We draw M samples from the prediction distribution (14), $\{\mathbf{u}_t^i\}_{i=1\ldots M}$, and then compute weights $w_t^i = \kappa\, p(\mathbf{z}_t \,|\, \mathbf{s}_t = \mathbf{u}_t^i)$, where $\kappa^{-1} = \sum_i p(\mathbf{z}_t \,|\, \mathbf{s}_t = \mathbf{u}_t^i)$. In this way, the samples are properly weighted samples from $\mathcal{P}_t$. Resampling then yields M equally weighted initial states. The HMC filter is therefore like a particle filter, beginning each time step with M particles, but each particle then spawns a Markov chain that converges to the target posterior. Fig. 5 depicts this convergence.

To obtain samples from the target posterior $\mathcal{P}(\mathbf{s})$, hybrid Monte Carlo performs a physical simulation of an energy-conserving system with a potential energy bowl equal to $-\log \mathcal{P}(\mathbf{s})$ [8, 19]. The intuition is that if you observe the state of the system at regular intervals, then the collection of observed states forms a Markov chain that comes from $\mathcal{P}$, provided you replace the system's momentum after every observation by a random Gaussian draw. The momentum resamplings ensure that the system can acquire enough energy to visit unlikely states with nonzero probability.

In the physical simulation, each state variable s_j is paired with a momentum variable p_j. On this extended state space, the Hamiltonian is defined as $H(\mathbf{s}, \mathbf{p}) = E(\mathbf{s}) + K(\mathbf{p})$, where $E(\mathbf{s}) = -\log \mathcal{P}(\mathbf{s})$ is the potential energy and $K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathbf{M}\mathbf{p}$ is the kinetic energy for a system with a diagonal mass matrix $\mathbf{M}$. The target distribution is then de-

fined as

$$\mathcal{P}'(\mathbf{s},\mathbf{p}) = C\exp(-H(\mathbf{s},\mathbf{p}))\ , \tag{15}$$

where C is a normalizing constant. By construction $\mathcal{P}'$ is separable, so the marginal distribution of $\mathbf{s}$ under $\mathcal{P}'$ is simply the desired posterior $\mathcal{P}$. Thus, if we were to draw sample $(\mathbf{s},\mathbf{p})$ from $\mathcal{P}'$, then $\mathbf{s}$ would be a fair sample from $\mathcal{P}$.

Hybrid Monte Carlo produces MC samples with a transition $(\mathbf{s}^r,\mathbf{p}^r) \rightarrow (\mathbf{s}^{r+1},\mathbf{p}^{r+1})$ that leaves $\mathcal{P}'$ invariant. (In what follows, we drop the superscript c and the subscript t with the understanding that the discussion applies to chain c at time t.) The HMC transition is composed of two steps, each of which leaves $\mathcal{P}'$ invariant. First, $\mathbf{p}^r$ is replaced by $\tilde{\mathbf{p}}^r$, sampled from a mean-zero Gaussian with covariance $\mathbf{M}^{-1}$. This leaves $\mathcal{P}'$ invariant as $\mathbf{p}$ is independent of $\mathbf{s}$, and we have not changed $\mathbf{p}$'s marginal distribution.

The second step, $(\mathbf{s}^r,\tilde{\mathbf{p}}^r) \rightarrow (\mathbf{s}^{r+1},\mathbf{p}^{r+1})$, involves the physical simulation. Starting from $(\mathbf{s}^r,\tilde{\mathbf{p}}^r)$, the system evolves according to Hamiltonian dynamics:

$$\frac{d\mathbf{p}}{dt} \ = \ -\nabla E(\mathbf{s})\ , \qquad \frac{d\mathbf{s}}{dt} \ = \ \mathbf{M}\mathbf{p}\ . \tag{16}$$

Because Hamiltonian dynamics conserves H, is reversible, and preserves the phase space volume, it leaves $\mathcal{P}'$ invariant [19]. In practice, however, the Hamiltonian simulation is performed numerically, in an iterative manner with a finite step-size (we use a sequence of deterministic leapfrog steps called a leapfrog trajectory [5, 8, 19]). As a result, the simulation is not guaranteed to conserve H exactly and leave $\mathcal{P}'$ invariant. To ensure that the Markov chain has the correct stationary distribution, we therefore perform a Metropolis rejection test to the state, $(\mathbf{s}^*,\mathbf{p}^*)$, at the end of each leapfrog trajectory [8, 18]; i.e., we accept $(\mathbf{s}^*,\mathbf{p}^*)$ with probability

$$\min\{1,\exp[-H(\mathbf{s}^*,\mathbf{p}^*)+H(\mathbf{s}^r,\tilde{\mathbf{p}}^r)]\}\ . \tag{17}$$

If accepted, we set $(\mathbf{s}^{r+1},\mathbf{p}^{r+1})$ to be $(\mathbf{s}^*,\mathbf{p}^*)$. Otherwise, it is set to the value of $(\mathbf{s}^r,\tilde{\mathbf{p}}^r)$.

The Metropolis test yields transitions with a stationary distribution if used with deterministic proposals that are self-inverting and have Jacobian 1. Our physical simulation has both properties. Although other types of proposals can be used with Metropolis tests to obtain samples from $\mathcal{P}'$, the key advantage of the Hamiltonian simulation is that H remains roughly constant even for long trajectories. One can see from (17) that keeping H roughly constant keeps rejection rates, and thus Markov chain autocorrelations low. Furthermore, long trajectories avoid random walks, and therefore produce samples efficiently from distributions [19].

Finding suitable step-sizes for the Hamiltonian simulation is important. If they are too small then the acceptance rates are high but we explore the space slowly relative to the amount of computation. If too big, then H may diverge and the rejection rates increase. Ideally, the step-size in each direction should scale with the width of the energy bowl in that direction. Based on Gaussian target distributions, Neal [19] suggests that the step-size for each state variable should be close to one standard deviation of the corresponding target marginal. One can show that this is achieved by setting the elements of the diagonal mass matrix $\mathbf{M}$ as follows:

$$\mathbf{M} \ = \ \mathrm{diag}(\epsilon_1^2,...,\epsilon_d^2)\ , \quad \text{where} \quad \epsilon_k = \left(\frac{\partial^2 E}{\partial s_k^2}\right)^{-\frac{1}{2}}, \tag{18}$$

where d is the state space dimension. Remember that the mass matrix may depend on the initial state $\mathbf{s}_0$, but any other dependence of the step-sizes on $\mathbf{s}$ would violate the self-inverting property of the transition. Therefore to find ϵ_k we compute derivative approximations using only $\mathbf{s}_0$.

The second derivative of the log posterior in (18) is equal to the sum of the derivatives of the log prior and the log likelihood. We approximate the log prior derivative using the variance of the process noise in the temporal dynamics. In deriving an approximation to the log likelihood derivative, we note that the phase observation density tends to dominate the shape of the likelihood surface. We therefore consider only the Gaussian component of the phase likelihood (5). Under this simplified model, suppose we have phase measurements at the mid-points $\mathbf{x}_j(\mathbf{s}_0)$ of each edge on the J body parts, and formulate the joint likelihood as

$$\log L(\mathbf{s}) \ \approx \ \log\prod_{j=1}^{2J}\left(\kappa\exp\left\{\frac{-(\phi(\mathbf{x}_j(\mathbf{s}))-\mu)^2}{2\sigma^2}\right\}\right)^{\frac{1}{n_j}} \tag{19}$$

$$= \ \sum_{j=1}^{2J}\frac{1}{n_j}\left\{\log\kappa \ - \ \frac{(\phi(\mathbf{x}_j(\mathbf{s}))-\mu)^2}{2\sigma^2}\right\}\ , \tag{20}$$

where κ is the Gaussian normalization constant.

To complete the approximation we exploit the pseudo-linearity of phase [9], and we adopt a first-order model for the spatial dependence of edge mid-points on the state, $\mathbf{x}(\mathbf{s})$. Together, these approximations yield a simple approximation for $\phi(\mathbf{x}_j(\mathbf{s}))$ in the neighborhood of $\mathbf{s}_0$, i.e.,

$$\phi(\mathbf{x}_j(\mathbf{s})) \ \approx \ \phi_0 + \nabla\phi^T\frac{\partial\mathbf{x}}{\partial\mathbf{s}}(\mathbf{s}-\mathbf{s}_0)\ , \tag{21}$$

where ϕ_0 is a constant independent of $\mathbf{s}$ and $\nabla\phi$ is the spatial phase gradient. Finally, we substitute this linear approximation into (20), and take the second derivative with respect to the state $\mathbf{s}$ to obtain:

$$-\frac{\partial^2\log L(\mathbf{s}_0)}{\partial^2 s_k} \ \approx \ \frac{1}{\sigma^2}\sum_{j=1}^{2J}\frac{1}{n_j}\left(\nabla\phi^T\frac{\partial\mathbf{x}_j(\mathbf{s}_0)}{\partial s_k}\right)^2\ . \tag{22}$$

To complete the estimation of ϵ_k in $\mathbf{M}$, we approximate the phase gradient, $\nabla\phi$, by the tuning frequency of the filter [9], and we approximate the position gradients by

$$\frac{\partial\mathbf{x}_j(\mathbf{s}_0)}{\partial s_k} \ \approx \ \frac{\mathbf{x}_j(\mathbf{s}_0+\triangle s_k)-\mathbf{x}_j(\mathbf{s}_0)}{\triangle s_k}\ . \tag{23}$$

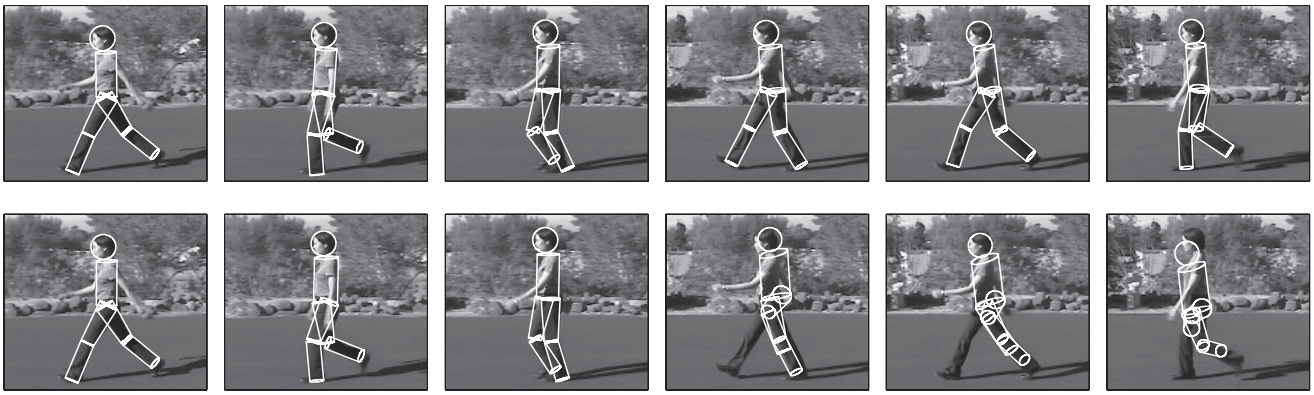

Figure 6. Cropped images showing every 4th frame (left-to-right) of 14D lower body trackers through self-occlusions. (top) HMC filter; (bottom) particle filter. The same computation time was used by both filters (about 3 min/frame on a 750 MHz processor). The HMC parameters were $M = 1$, $R = 200$, $b = 50$, $L = 60$, and $\epsilon = 0.2$. The particle filter used 20,000 particles.

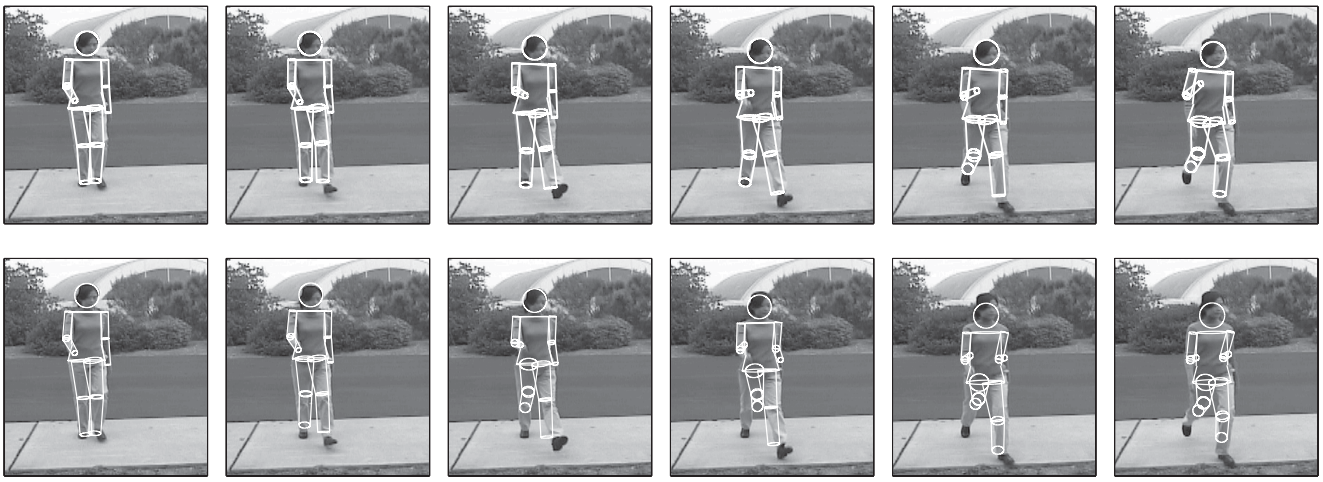

Figure 7. Every 2nd frame (left-to-right) of results from a 23D full body tracker, for the HMC filter (top) and for the particle filter (bottom). The same computation time was used by both filters (about 7 min/frame on a 750 MHz processor). The HMC parameters were $M = 10$, $R = 50$, $b = 30$, $L = 20$, and $\epsilon = 0.1$. The particle filter used 36,000 particles.

5 Experiments

We have used the HMC filter for tracking people in cluttered outdoor environments from grayscale, monocular video taken with an uncalibrated camera. We manually set the initial state at the first frame, and we roughly estimate the body dimensions and the intrinsic camera parameters by hand.

5.1 HMC and Particle Filter Parameters

While the particle filter has one main parameter (i.e., the number of particles) the HMC filter has several parameters. These include the number of chains M, the chain length R, the leapfrog trajectory length L, a stepsize adjustment factor ϵ, and, b, the number of burn-in samples at the beginning of each chain that are discarded. Both b and R depend on the convergence diagnostics of the Markov chain. While heuristic methods to detect convergence exist, one cannot determine when one reaches equilibrium with certainty.

To set these parameters we observed the samples from a long Markov chain in the first 3 frames. We then set b to the number of samples after which the posterior appears to have stabilized. We then set R so that we would obtain a sufficiently large sample set after equilibrium. The stepsize factor, ϵ, is used with $\mathbf{M}$ to control the Hamiltonian simulation step-sizes. Although $\epsilon \approx 1$ should be close to optimal given the approximate mass matrix in (18), for high-dimensional problems with correlated variables like the problems considered here a smaller ϵ is often preferred [19]. For small step-sizes more leapfrog steps are needed to avoid random walks, therefore L depends on ϵ. Based on HMC samples over the first 3 frames, we set L to keep autocorrelations be-

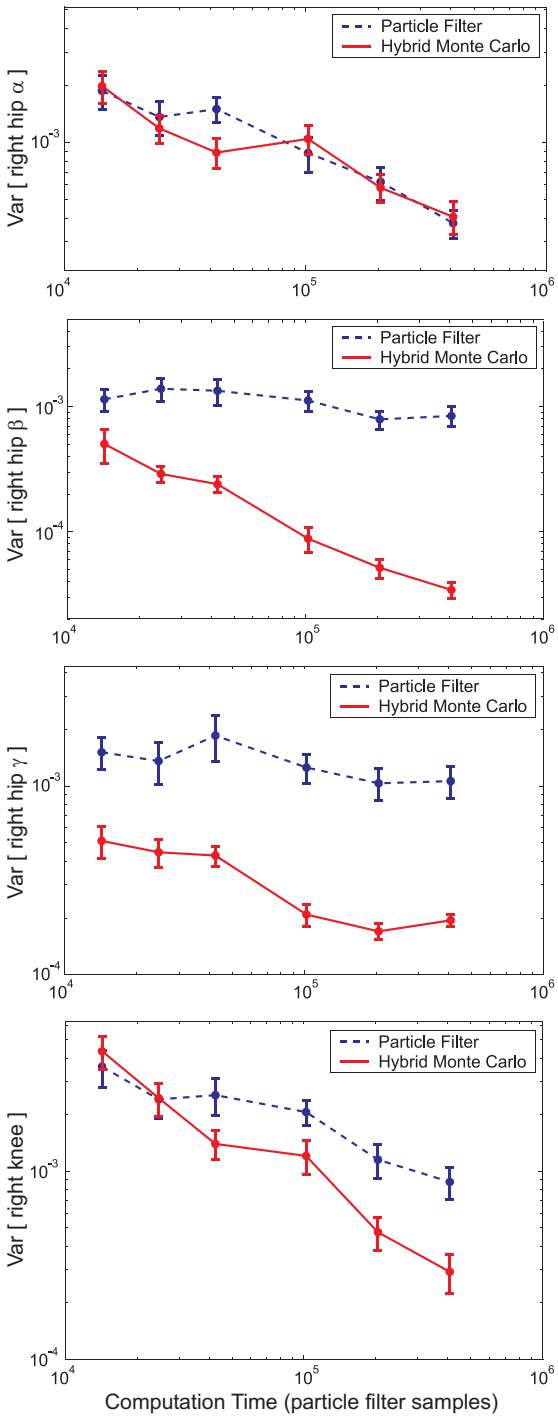

Figure 8. Estimator variance as a function of computation time for the 3-DOF right hip, and the right knee. Time is expressed in terms of the computation required by a single particle of the particle filter. Variances are based on 50 independent runs. Vertical bars indicate one standard error.

tween adjacent samples low. The specific parameters used are specified in figure captions associated with the results.

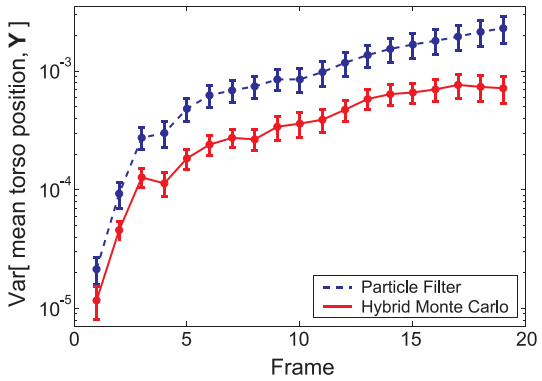

Figure 9. Estimator variance for estimates of the mean torso position over a 20-frame sequence. The vertical bars indicate one standard error, obtained from 40 independent runs of the filter.

5.2 Experimental Results

Fig. 6 shows the mean posterior state at every fourth frame for a typical run of the HMC and particle filters. This case illustrates results of tracking a 14D lower body model. It is easy to see that HMC filter yields better state estimates, especially in the presence of the self-occlusion of the legs. After the Markov chains reach equilibrium, HMC maintains a compact sample representation around the posterior mean. The average rejection rate in our HMC simulations was about 70%. This is higher than one might like, and certainly higher than that reported in [5]. We attribute the higher rejection rate to the more complex, noisy likelihood function associated with real images.

Similar results are shown in Fig. 7. In this case the subject is walking toward the camera, thereby undergoing scale changes. The human body model used here had 23 dimensions that also included arms and a nonlinearly tapered cylinder for the torso. Again, we find the HMC filter produces more accurate estimates of the body pose.

In comparing the two filter-based trackers it is also of interest to examine estimator variability (4). This is important since a reliable stochastic algorithm will generally produce reliable results when it is applied to the same data multiple times. Smaller deviations from ground truth are preferred. Furthermore, with an analysis of estimator variance one can hold variance fixed, and ask what computation time would be required to achieve such a level of confidence in the estimator.

Towards this end Fig. 8 shows examples of the estimator variances for mean state estimates for individual state variables. The image sequence was the same as that used in Fig. 6. The estimator variance was computed as the mean squared deviation from the ground truth mean state. The ground truth mean state was found using a particle filter with 4×10^6 particles, many more than used in the estimator variance experiments. The mean squared estimates are com-

puted from mean state estimates obtained in 50 independent runs of each filter. Fig. 8 shows estimator variances for the HMC filter and the particle filter as a function of computation time (measured in terms of the computation time required by a single particle of the particle filter). Here we show results for the 4 state variables of the right leg, including the 3 degrees of freedom of the hip, and the knee angle.

These figures are typical of the types of results we have observed across different runs and different points in the gate cycle. The estimator variance of the HMC filter is usually 5-20 times smaller than that of the particle filter. Moreover, in the range of computation times shown here the difference between the variance typically grows as the available computation time increases. The effective number of samples for the particle filter remains small until one typically has several hundred thousand particles. The result shown in the top panel of Fig. 8 in which the HMC filter and the particle filter perform similarly is typical of only a very small number of variables on different runs.

Finally, it is also of interest to consider how the estimator variances change as a function of time. Toward this end, Fig. 9 shows the mean state variance for a torso positional variable, over 20 frames of the sequence used in Fig. 6. Again it is clear that the HMC filter has a lower estimator variance in general.

6 Discussion and Future Work

This paper describes the use of hybrid Monte Carlo for inferring the 3D shape and motion of people from monocular video. The HMC filter is shown to be more effective for high-dimensional Bayesian filtering problems than a conventional particle filer. Nevertheless, there remain many avenues for future work, including the formulation of better prediction distributions and initial states for HMC Markov chains, the integration of edge and motion information for improved likelihoods, and adaptive tuning of HMC parameters to improve convergence and mixing.

Acknowledgements: The authors thank Kiam Choo and Radford Neal for their comments on this research. EP thanks NSERC Canada and the Xerox Foundation for their financial support.

References

[1] M.J. Black and D.J. Fleet. Probabilistic detection and tracking of motion discontinuities. *IJCV*, 38:229–243, 2000

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

[3] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proc.-Radar, Sonar Navig.*, 146:2–7, 1999

[4] T. Cham and J.M. Rehg. A multiple hypothesis approach to figure tracking. Proc. IEEE CVPR-98, Fort Collins, V. II, pp. 239–245

[5] K. Choo and D. J. Fleet. Tracking people using hybrid Monte Carlo. Proc IEEE ICCV-01, Vancouver, V. II, pp. 321–328

[6] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealled particle filtering. Proc IEEE CVPR-00, Hilton Head, V. II pp. 126–133

[7] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, Berlin, 2001

[8] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195:216–222, 1987

[9] D. J. Fleet and A.D. Jepson. Stability of phase information. *IEEE Trans PAMI*, 15:1253–1268, 1993.

[10] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Trans PAMI*, 13(9):891–906, 1991.

[11] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. Radar, Sonar and Navig.*, 140:107–113, 1993

[12] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29:2–28, 1998

[13] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. Proc ECCV-98, Freiburg, V. I, pp. 893–908

[14] M. Leventon and W. Freeman. Bayesian estimation of 3D human motion from an image sequence. MERL TR-98-06

[15] J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In A. Doucet et al, *Sequential Monte Carlo Methods in Practice*. Springer, 2001

[16] J.S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *JASA*, 93:1032–1044, 1998

[17] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. Proc ECCV-00, Dublin, V. II, pp. 3–19

[18] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953

[19] R.M. Neal. *Bayesian Learning for Neural Networks*. Springer, NY, 1996. Lecture Notes in Stats., No. 118

[20] O. Nestares and D. J. Fleet. Detection and tracking of motion boundaries. Proc IEEE CVPR-01, Kauai, V. II, pp. 358–365

[21] R Plankers and P. Fua. Articulated soft objects for video-based body modeling. Proc IEEE ICCV-01, Vancouver, V. II, pp. 394–401

[22] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. Proc ECCV-00, Dublin, V. II, pp. 702–718

[23] C. Sminchisescu and B. Triggs. Hyperdynamic importance sampling. Proc ECCV-02, Copenhagen, V. I, pp. 769–783, Springer