SHAPE AND MOTION DRIVEN PARTICLE FILTERING FOR HUMAN BODY TRACKING

Takashi Yamamoto and Rama Chellappa

Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742-3275

ABSTRACT

In this paper, we propose a method to recover 3D human body motion from a video acquired by a single static camera. In order to estimate the complex state distribution of a human body, we adopt the particle filtering framework. We represent the human body using several layers of representation and compose the whole body step by step. In this way, more effective particles are generated and ineffective particles are removed as we process each layer. In order to deal with the rotational motion, the frequency of rotation is obtained using a preprocessing operation. In the preprocessing step, the variance of the motion field at each image is computed, and the frequency of rotation is estimated. The estimated frequency is used for the state update in the algorithm. We successfully track the movement of figure skaters in TV broadcast image sequence, and recover the 3D shape and motion of the skater.

1. INTRODUCTION

The task of human body tracking using a video is to identify and locate every body parts frame by frame. This is an essential process for automatic interpretation of human activities. Applications include human identification, virtual reality visualization, wireless user interface for video games, and motion analysis in sports and medicine.

Unlike a single rigid object, human body tracking is a challenging problem due to the complexity of motion parameter space and self-occlusion. Many approaches define 3D human model by ellipsoids, spheres, and cylinders to approximate the body configuration [1, 7, 8]. By imposing the joint constraints, the model can exclude many translation parameters from consideration. Breger et al suggests another way to describe the human motion using twist [1]. This concept originally comes from the literature in robot manipulation. By expressing rotational parameters with twist, the estimation of complex motion parameters is simplified. Some extended works [11, 2] have also used color and texture information for more robust tracking.

Our algorithm is an extension of the particle filtering method [4, 5, 6] with a 3D model. Instead of tracking one rigid object, we need to handle an articulated object in an organized way. We attack this problem using two approaches.

- Effective parameter search

The particle filtering approach has a drawback in handling the number of particles for complex/multiple objects. Usually, human body tracking with a simple movement requires 20000 samples per frame to get reasonable results [9]. Partitioned sampling offers an elegant way for reducing the number of samples, still providing a good approximation to the state distribution [6].

- Use prior knowledge on motion

To improve performance, we obtain prior knowledge of motion using a preprocessing stage. For example, if we know in advance that a person walks from left to right, we move fewer particles to the left in the next frame and put more particles towards the right. Examples of utilizing prior information to improve the state estimate are in [9, 10]. Motivated by this idea, first we preprocess the image sequence to extract useful information about rotation, then use it to achieve improved performance.

In the implementation, first, the human body model is decomposed into several layers so that the state estimate can be refined as we process each layer. In order to deal with the rotational motion, we obtain the frequency of rotation from the motion field. Then this motion cue is fused into the algorithm to improve the tracking performance.

2. ALGORITHM

2.1. Model

Partitioned sampling [6] effectively produces particles that are useful for the state estimate. We consider the human body as a tree structure and decompose it into several layers so that the partitioned sampling operation can be exploited at each layer. For simplicity, suppose that we track an arm. The arm can be divided into two layers: the upper arm, and the lower arm layer. Each body part has its own local coordinate system attached to it. Note that we use the hierarchical structure so that the local coordinates of the lower arm is relative to the local coordinates of the upper arm. Let \([x, y, z]\) and \([r, x, y, z]\) denote the translation and rotation parameters for the x, y, and z axis respectively. The state vector of the arm at time \(t\) can be expressed as \(X_t = [u_{at}, l_{at}]^T\), where \(u_{at} = [t_{ax}, t_{ay}, t_{az}, r_{ax}, r_{ay}, r_{az}]^T\), and \(l_{at} = [r_{lx}, r_{ly}, r_{lz}]^T\). From the joint constraint, we do not need to specify the translation parameters for \(l_{at}\) since the lower arm cannot translate relative to the local coordinate of the upper arm.

Assuming that the dynamics of the upper and the lower arm are independent each other, we update the state \(X_{t-1}\) to \(X_t\) as

\[
\begin{pmatrix}
    u_{at-1} \\
    l_{at-1}
\end{pmatrix} \overset{p(u_{at}|u_{at-1})}{\sim} \begin{pmatrix}
    u_{at} \\
    l_{at}
\end{pmatrix} \overset{p(l_{at}|u_{at})}{\sim} \begin{pmatrix}
    u_{at} \\
    l_{at}
\end{pmatrix}
\]

This diagram describes the two step dynamics: apply the dynamics for the upper arm only, then apply the dynamics for the lower arm. This arm example can be readily extended to more general models.
2.2. Particle filtering

The particle filtering is a sequential Bayesian estimation algorithm for the posterior distribution of a state vector. The Kalman filter optimally estimates the position when the state distribution is governed by Gaussian density and the system is linear. In many cases, however, due to background clutter, self-occlusion and shape deformation, the density of the state space becomes non-Gaussian. On the other hand, particle filtering can handle complex distributions by sequentially approximating the posterior distribution with a set of particles.

Let $X_0, \ldots, X_t$ denote a dynamical state process. This process is modeled as a Markov process with initial distribution $p(X_0)$ and a transition probability $p(X_t | X_{t-1})$. Let $Z_1, \ldots, Z_t$ denote the observation process, and they are assumed to be mutually independent given the dynamical state process.

In the sampling stage, a set of state particles $(x_i^{(t)}, \pi_i^{(t)})_{i=1}^N$ is randomly drawn according to the prior distribution $p(X_{t-1} | Z_1^{t-1})$, and all weights are set to $\pi_i^{(t)} = 1/N$.

Then we apply the dynamics process $p(X_t | X_{t-1})$, which is summarized as

$$X_t = X_{t-1} + \text{drift}_{t-1} + \text{diffusion}_{t-1}$$

So, every particle (state) is moved by the same amount $\text{drift}_t$ and then each particle is randomly perturbed by $\text{diffusion}_t$. Usually, this object dynamics is expressed as a second order process [4, 5].

In the third stage, we take a similarity measurement between each state and the input image. This step corresponds to re-weighting the particle weight $\pi_i^{(t)}$ with $p(Z_t | X_t = x_i^{(t)})$. After normalizing each weight, we obtain the approximated posterior distribution from the particles.

Finally, the mean of the current state can be computed by the updated particle set as

$$\bar{X}_t = \frac{1}{N} \sum_{i=1}^N x_i^{(t)} \pi_i^{(t)}$$

Now, the weighted resampling operation, which is explained in the next subsection, is inserted between the decomposed dynamics. In this way, more effective particles are generated at the desired state and ineffective particles are removed as we process each layer.

2.3. Weighted resampling

The weighted resampling is a sampling operation applied to a set of particles without changing its representation as a probability distribution [6]. Let $(x_i, \pi_i)_{i=1}^N$ be a particle set of states $x_i$ and weight $\pi_i$. This particle set represents $p(x)$. And let $(q_j)_{j=1}^M$ be a list of weights representing a strictly positive function $g(x)$ on the support of $p(x)$, then

1. For $i = 1, \ldots, n$,
   1. With probability $q_j$, pick an index $j$. (resample)
   2. Set $x'_i = x_j$.
   3. Set $\pi'_i = \frac{\pi_j}{q_j}$. (reweight)
   2. Normalize $\pi'_i$ such that $\sum_{i=1}^N (\pi'_i) = 1$.

From the resampling operation, we expect that many particles are generated at the peak of the importance function $g(x)$. Though the representation of the distribution is heavily distorted by the importance function at this point, the reweighting process neutralizes it back by putting less weight to the particles drawn from the peak of $g(x)$.

Since the weighted resampling does not change the representation of a probability distribution, we can put this operation after each decomposed dynamics. The motivation to insert the weighted resampling is to generate more particles at the promising state, assuming the peak of the importance function is closer to the true state. So, from the arm example, one appropriate choice of the importance function is the local measurement function of the upper arm

$$g_{\alpha} = p(Z_t^{\alpha} | u_{\alpha}, \alpha_{t-1})$$

where $Z_t^{\alpha}$ denotes the observation of the upper arm at time $t$. The algorithm is summarized in figure 1.

The measurement function is constructed according to a measure of similarity in shape between the observation and model. Details are fully described in [7].

3. IMPLEMENTATION

Our 3D human model defines ten body parts and is represented using five layers in a hierarchical structure: (head and torso), (right upper arm and lower arm), (right thigh and calf), (left upper arm and lower arm), (left thigh and calf). The head motion is defined using six parameters $[tx, ty, tz]^T$, $[rx, ry, rz]^T$, and the rest of the body parts define only rotation parameters $[rx, ry, rz]^T$ due to the joint constraint. We pick 700 particles for each layer. Initialization is done by a careful initialization of the body position so that the initial state distribution $p(X_0)$ is defined.

Image data with a single camera is the input observation. Our image data are real TV broadcast figure skating image sequences. Each sequence has only one skater. In a realistic situation, the camera is not stationary and therefore the background also moves,
making shape-encoded measurements almost ineffective when the background has the similar shapes as body parts. In the experiment, we use a static scene segment so that we can obtain reasonably good background subtracted images.

In order to track a rotating human, frequency of the rotation is obtained using a preprocessing operation. In preprocessing, the variance of the motion field at each image is computed, and then the detected periodicity is used to provide a clue to frequency estimation. The estimated frequency is used for the state update in the algorithm.

3.1. Preprocessing

Several assumptions are made in the preprocessing stage to estimate the frequency of rotation. They are: (1) In the input image sequence, the action of a person is only rotation. (2) S/he is spinning along the y-axis of the reference point. (3) The initial pose of the person is known. The reference point is the coordinate origin of the first body part in the tree structure. In our implementation, the origin of the head coordinate is the reference point. Suppose we process $N$ frames. Denote a histogram of gradient magnitude at $T=t$ as $H_t$. Then

1. Compute the histograms of gradient magnitudes for all the image frames and take the segment $H_{t}, H_{t+1}, ..., H_{t+N-1}$.
2. From the histograms obtained in step 1, compute the variance of the process $r(t) = Var(H_t)$.
3. Take the discrete Fourier transform of the process $r(t)$ and obtain the most dominant frequency $f_0$. This is the frequency of half rotation.

Due to the symmetry of the human body structure, during a rotation, we observe two periodic pattern. The updates of dynamics of the reference point is summarized as follows.

1. Obtain the angle of rotation around the y-axis $\text{rot}_Y$ by
   \[ \theta_y = \frac{r_0}{f_0} \]
2. For $\text{rot}_Y$ component of the reference point, use the update
   \[ \text{rot}_{Y'} = \text{rot}_{Y'-1} + \theta_y + w_{z-1} \]
   For other components, use
   \[ X_t = X_{t-1} + d_t + w_t \]

where $d_t$ and $w_t$ denote the deterministic drift and random perturbation respectively.

The idea is, first, using the preprocessing operation, we estimate how much the person rotated from one frame to the next. Second, we estimate the motion of each body part using particle filtering.

4. RESULT

For the 'Sarah' sequence, with the hand-initialization for the first frame, we tracked the skater's movement over 20 image frames. Although the skater’s right leg is heavily self-occluded in the first few frames, the algorithm could catch its movement later when it becomes visible. See figure 2.

In figure 2, we see the advantage of having a 3D body model. Once the full 3D information of the movement is recovered, we can see the reconstructed body movement from every possible view point. The third column of figure 2 shows the skater's movement from the left side view.

Another experiment was done on the image sequence of a rotating skater. For 'Liu' sequence, we preprocessed the image frames using a moving window with 50 frames. In addition, we selected the image sequence of the rotating skater so that the action of the skate is known in advance.

We tracked one complete rotation of the skater using our method. See figure 3 for the results. In 'Liu' sequence, it takes about 23 frames for the skater to complete one rotation. The third column of figure 3 shows the reconstructed movement seen from the ceiling. Though we observe distortions especially in the right leg, reasonably good body visualization was achieved.

4.1. Discussion

Suppose we have the same number of particles for both layer and non-layer structured particle filtering algorithms. In the latter case, the particles are distributed in a large area and most of the particles are killed whereas the layered algorithm has more compact distribution with many effective particles. Therefore, the ratio of useful particles to the whole particle set turns out to be different for the two methods [6]. The comparison of these two approaches can be seen in figure 4. Direct implementation in (a) shows a large variation of particle states while layered implementation (b) shows a compact distribution. For illustration purposes, states of particles whose weights are under a predefined threshold are not displayed.

Theoretically speaking, we should decompose the body model into as many layers as possible to obtain proportionally many effective particles to the entire particle set. But in practice, this ultimate choice may introduce errors since human motion almost always causes self-occlusion. When occlusion happens, the approximation of likelihood function $p(Z|X)$ is quite poor since the measurement becomes highly biased or even impossible. Therefore, if we apply the weighted resampling to the layer whose body part is invisible, most of particles are killed resulting in poor approximation of the posterior distribution.

If such a poor approximation occurs somewhere in between
the decomposed dynamic process, rest of the approximation collapses while the straightforward implementation is more robust since other visible body parts can be incorporated into the observation. It is important to put the most reliable body part into the first layer. In most cases, the head is visible and easy to locate in an image sequence [9]. And we should put a few body parts in each layer to be able to observe visible body parts all the time. This avoids potential damage due to occlusion. For example, in case the right thigh is occluded by body parts, we may still observe the right calf to make a measurement if thigh and calf are in the same layer.

During rotation for the 'Liu' sequence, we observe a large amount of occlusion. In this experiment, we let a completely occluded body part remain as the same position as in the previous frame. Unlike independent objects, the occluded body part is dragged by other body parts because our 3D human model has joint constraints. Therefore, leaving them in the same state as the previous frame is not a bad choice. If there is a situation that a body part is only partially occluded, then it is possible to make a shape filter for only the visible parts.

5. CONCLUSION

Object tracking driven by particle filtering is extended to human body tracking algorithm. The large number of parameters is handled by decomposing the body parts into several layers. In addition, in order to deal with the rotational motion, the frequency of rotation is obtained using a preprocessing operation.

6. REFERENCES


