Bayesian 3D Human Motion Capture Using Factored Particle Filtering

Abdallah Dib†, Cédric Rose‡, François Charpillet†

†Nancy Université / ‡INRIA / †DIATELIC S.A.
LORIA – Campus Scientifique – BP 239 / 54506 Vandœuvre-lès-Nancy Cedex – France
Email: firstname.lastname@loria.fr – http://maia.loria.fr

Abstract—We present a markerless human motion capture system that estimates the 3D positions of the body joints over time. The system uses a dynamic bayesian network and a factored particle filtering algorithm. In this paper we evaluate the impact of using different observation functions for the bayesian state estimation: chamfer distance, a pixel intersection intersection and finally a pseudo-observation of the subject direction calculated from the previous output of the system. We also compare two methods for the factored generation of the particles. The first one uses a deterministic interval exploration strategy whereas the second one is based on an adaptive diffusion. The capacity of the system to recover after occlusion by obstacles was tested on simulated movements in a virtual scene.

I. INTRODUCTION

Several strategies have been proposed for realizing 3D motion capture from video cameras. Some using sets of predefined patterns, other using an articulated body model. In both cases the problem can be formalized as a search in the space of the possible configurations of the human body. Gavrila et al. [1] proposed a hierarchic search in a set of templates using a distance measure based on a distance map constructed from edge detection. Deutscher et al. [2] presented a 3D motion capture algorithm based on an articulated model and a particle filtering algorithm. Saboune et al. [3] presented a human motion tracking system based on a modified particle filtering algorithm using a deterministic exploration of parts of the state space. Rose et al. [4] reduced the complexity of the particle filtering algorithm by formalizing the tracking problem as a dynamic bayesian network inference problem, taking advantage of the factored representation of the state space.

In this paper we present a 3D human motion capture system which is intended to be used for the monitoring of elderly people at home. We start from the formalized factored approach presented in [4] and we study two particle diffusion processes. We analyze the impact of merging silhouette information with edge information in the particle filtering algorithm. We also introduce a pseudo-observation concerning the orientation of the body. Finally we present an experimental analysis for evaluating the gain from merging the different observations and modifying the diffusion process.

II. BACKGROUND

A. Dynamic Bayesian network for human motion capture

Bayesian networks (BN) are a graphical formalism for reasoning about independence between variables. When working with a time-evolving set of variables $X_t$, we call the BN is called a Dynamic Bayesian Network (DBN)[5]. The state of a set of articulated segments such as the human body find a natural representation as a DBN which gives the advantage of reducing inference complexity [4]. Particle filtering can be used to estimate a posterior probabilities in a DBN given a sequence of observations [5].

The kinematic structure of the human body can be represented in a factored manner by considering the torso position at the root of a tree structure composed by the different body parts. A representation of the DBN is shown on figure 1. The dynamic constraints used are those of the human body joints.

B. Chamfer distance

The chamfer distance let us exploit the edges as additional information available in the input image.

A distance map [6] is created from the edge image and matching is done by translating the template at various locations of the distance image.

III. APPROACH

We used a 3D articulated body model to simulate the human movements through the configuration of 8 degrees of freedom. These degrees of freedom represent rotations.
of 5 joints of the body and $x, y$ and $z$ position of the torso. We defined three likelihood functions to evaluate each configuration. This functions are: silhouette image (labeled $P_t$ on figure 1), distance map (labeled $C_t$ on figure 1) and the last one based on the previous system output (labeled $O$ on figure 1). For the first two functions, the 2D projection of the 3D configuration is obtained and compared to the silhouette and distance map images. This comparison can be made in a factored manner by projecting the body parts sequentially on the silhouette and distance map respectively. The pseudo-observation is only used for evaluating the orientation of the torso as shown in figure 1. Next, after presenting each of the three likelihood function, Section III-D describes how this three functions will be combined in order to obtain a final weight for a given particle.

A. Silhouette observation

The silhouette of the real image is constructed by subtracting the background from the current image and then applying a threshold filter. The final score $P$ of a given particle will be the number of common pixels $I$ between the 2D projection of the configuration and the real silhouette divided by the number $N$ of pixels that are not common.

$$P = \frac{I}{N + 1}$$

B. Chamfer distance observation

A distance map is created from the edge image of the real image. Next, we perform a 2D projection for each body part configuration on the distance map. The particle’s score is calculated as follows:

$$C = \frac{1}{n} \times \sum_{i} d_{i}$$

where $i$ is the edge pixel of the evaluated configuration, and $d_{i}$ is the corresponding value on the distance map.

C. Pseudo observation of orientation

Since the system can recover the 3D position of the subject in movement, we will use this position found at previous time as a likelihood function for our particle filtering algorithm in order to determine the subject orientation. The final score of the particle is

$$O = -\theta^2_{t} \times \| P_{t-m}P_{t} \|$$

where $P_{t-m}P_{t}$ is the observed vector from previous positions and $\theta_{t}$ is the angle between the orientation of the particle and the vector.

D. Merging the three observations

We remind that when performing Bayesian inference we basically do the following calculation (bayes rule):

$$P(X|Scores) = P(Scores|X) \cdot \frac{P(X)}{P(Scores)}$$

where $P(X)$ is the prior estimated in the dynamic case by prediction from the previous state and $P(Scores)$ is the probability of the observation (given the model) and serves as a normalizing constant. Therefore, only the relative values of $P(Scores|X)$ for the different state configurations $x^j$ matter in the inference process. For a given particle $x^j$, observations are extracted from each of the camera images.

We simply use soft-max distributions for interpreting the score as probabilities. For each camera $k$ and each score $S$ (which can be $P$, $C$) we calculate:

$$P(S^k|x_i) = \frac{\exp \frac{S^k_i}{T_5}}{\sum_j \exp \frac{S^k_j}{T_5}}$$

where $S^k_i$ is the score of particle $i$ on camera $k$ and $T_5$ is a "temperature" constant predefined for each score type. The weight of the particles is then updated using:

$$w_i = w_i \times P(O|x_i) \prod_k P(P^k|x_i) \prod_k P(C^k|x_i)$$

E. Diffusion strategies

In this section we present the two diffusion strategies that we used in the particle filtering algorithm.

1. Static diffusion strategy: This strategy uses a deterministic interval exploration. The diffusion interval, calculated for each of the $N$ surviving particles from the dynamics constraints of body joints, is regularly sampled into $K$ points. Thus the total number of generated particles is equal to $N \times K$.

2. Adaptive diffusion strategy: This strategy uses an adaptive diffusion based on the parent weight. After particle selection, the number of generated particles at next iteration is calculated for each selected particle proportionally to its weight. This re-sampling process is the one used in the condensation algorithm [7].

IV. EXPERIMENTAL ANALYSIS

A. Experimental setup

We present results from three video captures of a subject in movement who walks freely in a scene. Before any processing, the cameras must be calibrated. The image processing stage and the particle’s weight calculation (pixel count and chamfer distance) was done on GPU using OpenGL shaders and CUDA.

B. Results

We reconstructed the 3D trajectory of the filmed subject using manual labeling of the body joints in the 2D images. This reconstructed trajectory was used as the reference when comparing different methods.

For the adaptive diffusion algorithm, the total number of generated particles is 50, while for the static diffusion algorithm the number was 200. Tables I and II show respectively the error on the torso and left ankle position of
the subject using different score functions with adaptive and static diffusions. In these tables, P stands for pixel count, C stands for chamfer distance and O stands for orientation. We denote as Ad the adaptive diffusion method and as Sd the static one.

The results in these tables show that the adaptive diffusion gives better results and it also speed up the search process by reducing the number of particles. The results also show that system is always able to locate the subject position in the 3D space. The pseudo-observation globally improves the results as can be seen for the ankle. Using the distance map in addition to silhouette observation improves the overall tracking process but has an additional computation cost of 15% in our implementation. Using the three observations with the adaptive diffusion method results in a more accurate tracking.

Table III shows the average and max error in cm for different body parts positions using the three likelihood functions with adaptive diffusion.

V. ROBUSTNESS AGAINST OCCLUSION

This section presents an experiment that we used for the validation of the algorithm. We developed a 3D Articulated human body model which was used for the generation of motion sequences recorded by several virtual cameras. As the system was designed for tracking persons in an interior environment with lot of obstacles, we need to test its capacity of recovering after occlusion. For that, we created a virtual scene with obstacles filmed by four virtual cameras, then we tested the two diffusion methods. We used the three likelihood functions for both cases.

Results show that with adaptive diffusion, the system was able to recover in 0.52 sec while with static diffusion it took 2.08 sec for the system to recover.

VI. CONCLUSION

We presented a 3D motion capture system based on a dynamic Bayesian networks and a factored particle filtering algorithm for taking advantage of the factorization of the state space. Compared to [4], we introduced two new observation functions in the particle filtering algorithm: chamfer distance, and pseudo-observation of the walk direction. We also compared two methods for the factored generation of the particles, the results show that the last strategy gives better results and reduces the number of required particles. Finally, we tested the capacity of the system to recover from occluded positions.

In this work, we were able to reduce the number of required particles without affecting the quality of the movement reconstruction.

REFERENCES