A Stereo Camera Based Full Body Human Motion Capture System Using a Partitioned Particle Filter

Zhenning Li and Dana Kulić

Abstract— **In this paper, we propose a marker-less full body human motion capture system designed for humanoid robot applications. The system is based on a stereo camera, and therefore has strong portability. Tracking is implemented within the particle filter framework, and the high dimensionality problem is solved through partitioned sampling. Taking advantage of the stereo setup, we propose a depth cue which resolves the problem of missing depth information in monocular tracking. Three other cues, the edge cue, the color cue and the distance cue, are also integrated into the system to enhance the tracking performance. The system is tested using the publicly available CMU MOCAP database which also includes ground truth data, and this enables us to analyze the results quantitatively and compare the relative usefulness of different cues. The system is shown to be capable of tracking challenging videos accurately and robustly in near real-time.**

I. INTRODUCTION

With the fast development of robotics and artificial intelligence technology, there has been a recent increase in research on humanoid robots. However, the issue of how to interact with the robots more comfortably and how to enable humanoids to accumulate motion knowledge autonomously remains an open problem. Similar to humans, robot vision could also be an effective sense to gather information and to learn from human teachers. Vision based human motion capture (human MOCAP) [1], [2] technology which is capable of real-time tracking is a promising way to solve these problems.

Commercial motion capture systems such as Vicon [3] are already available, however, most of these systems require multiple cameras permanently installed in a capture studio, and are based on markers, severely limiting their application. Consequently, in recent years more efforts have been devoted to the research of marker-less human MOCAP [4]. Model based tracking exploits the characteristics inherent in a human body. The typical model used is an articulated object model, but other models, such as the loose-limbed model, have also been proposed [5].

Many systems proposed in the literature consider the scenario with a multiple camera setup [6], [7], [5], [8], [9]. The cameras are separated by certain angles, and the human motion can be observed from different directions. In this approach, the problems of occlusion are alleviated remarkably. Although high accuracy can be achieved, the processing speed is usually far from real-time. Moreover, since the pre-calibrated cameras are mounted at permanent locations, this limits the tracking area and entails high hardware expenses.

Human MOCAP based on monocular video sequence is an alternative way [10], [11], [12], [13]. A single camera could be easily mounted to a robot head, but the missing depth information makes the tracking much more difficult. To incorporate depth information, several systems based on stereo camera have been developed [14], [15], [16]. In [15], Azad et al. propose an upper body tracking system, which uses the depth information obtained from the stereo camera implicitly. During the likelihood calculation, the predicted model is compared with both images of the stereo image pair, and the final weight is obtained by integrating the results from the two pipelines. However, their method will not function well when tracking motions which are complicated or include noticeable depth changes.

Marker-less human MOCAP is a very challenging task because the variety of possible human motions is very large, and the motions are almost random if the motion type is not specified. The complexity of the human motion makes the model dynamics nonlinear, while the complicated nature of the observation process causes the posterior density to be non-Gaussian. The particle filter [17] is well suited for this context. However, the number of particles required for successful tracking increases exponentially with the dimensionality, and this makes the basic particle filter quickly intractable for high Degree of Freedom (DOF) human models [18]. As a result, variants of the basic particle filter have been proposed, among which the best known are the annealed particle filter [6] and the partitioned particle filter [18].

In this paper, we propose a new articulated model-based full body human motion capture system intended for a humanoid application, which is capable of running in near real-time. The system is designed for stereo input, and the human motion is tracked by a particle filter with partitioned sampling in order to solve the high dimensionality problem. The tracker makes use of multiple cues, including a newly developed depth cue. To enable a quantitative error analysis of the algorithm performance, rather than using video obtained from a stereo camera, we use videos from the publicly available CMU MOCAP database [19], which also includes ground truth data obtained from a marker-based motion capture system. Since the database does not provide access to stereo camera images, we generate virtual depth images to simulate the true depth images. The system is tested with challenging videos, and the results demonstrate

The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

Z. Li and D. Kulić are with the Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, Canada z237li@uwaterloo.ca, dkulic@ece.uwaterloo.ca

that our system is capable of tracking through random, fast and complex motions with high speed and good accuracy.

II. Particle Filtering and Partitioned Sampling

The particle filter is the result of applying the Monte Carlo method in recursive Bayesian filtering [20], [21], which is used to estimate the posterior Probability Density Function (PDF) of the state variable based on Bayes' Theorem. When applied to model based human MOCAP, the state is a vector formed by all the joint angles and a root translation.

A typical recursive Bayesian filter has a two-step procedure, namely a prediction step and an update step. In the prediction step, the prior probability of the state at time step k is predicted according to the posterior probability of the state at time step $k-1$ and the prediction model. In the update step, the prior probability is updated into the posterior probability by incorporating the observation at time step k.

Under the assumption of a linear system with Gaussian noise, the optimal estimation is achieved in terms of the minimum covariance, i.e., the Kalman Filter [22]. In a particle filter (also known as the Condensation algorithm [17]), the PDF is sampled and represented by a set of particles with weights proportional to the likelihood. This representation of the distribution makes no assumption about the distribution shape, and is capable of handling non-Gaussian and multimodal distributions.

Computing the particle weights is a key issue. In the vision based tracking context, commonly used cues for the weighting computation are color [13], [14], edge [6], [15], [13], [8], [14], region [6], [9], [8], distance [15], [14] and motion [13].

When implementing a particle filter, the biggest problem for full body human tracking is the high dimensionality. The number of particles required for successful tracking increases exponentially with the increase in dimension. One approach for handling the issue of high dimensionality is the annealed particle filter [6], which tries to focus more particles to the global maxima, reducing the number of particles required. However, this leads to the loss of diversity, making it difficult for the tracker to recover if the target is lost. This motivates us to use the partitioned particle filter.

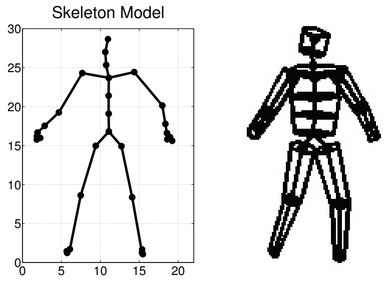
Partitioned sampling was first proposed by J. MacCormick et al. in [23]. The partitioned particle filter deals with the high dimensionality problem by breaking the high dimensional state space down into several subspaces. In this way, the number of particles required for successful tracking can be reduced.

In our application, the 25 DOF state space is divided into ten partitions: torso (6 DOF), head (3 DOF), two upper arms (3 DOF for each), two forearms (1 DOF for each), two thighs (3 DOF for each) and two calf legs (1 DOF for each). For each frame, the same set of particles are used in all the partitions, but with reproducing and perishing. This is different from tracking the partitions using independent filters. In the latter case, only the estimated position is passed down from the previous partition to the subsequent partition. For an articulated object, this will lead to accumulating errors from layer to layer. In the partitioned sampling approach, the estimated PDF is passed down. The advantage is that the particles maintain the estimated distribution of the previous partitions while evaluating the subsequent partitions. The final resampling and estimation is done according to the distribution obtained by combining the distributions of all the partitions. In this way, by storing multiple hypothesises of the preceding partitions, it is still possible to find the correct configuration even if the previous partitions cannot be well localized.

III. Particle Filter Implementation

A. Human Model and Projection Model

1) Human Model: The human model is composed of a skeleton model (Fig. 1.1) and an outer shape model (Fig. 1.2). The skeleton model contains 25 DOF: 3 DOF for the root translation, 3 DOF for the root rotation, 3DOF for the neck, 3 DOF for each shoulder, 1 DOF for each elbow, 3 DOF for each hip and 1 DOF for each knee. For each joint angle, an angle limit is specified to reduce the search space and to avoid impossible configurations.



1.1: Skeleton Model 1.2: Outer Shape Model

Fig. 1. Human Model

Moreover, an outer shape model is formed using truncated cones to describe the model surface. Each truncated cone is described by two circles and a mesh formed by points sampled from the cone surface. As more points are sampled, more computation will be required during the tracking. Therefore the sample density is made adjustable, so that we can balance the model accuracy and the computation time.

2) Projection Model: When weighting the particles, we need to compare the predicted configuration with the image data. The projection model, which describes how the camera will project the human model from 3D to 2D, is therefore required. Using the videos and the corresponding ground truth data of the human motions, we obtained the projection model through nonlinear fitting. Through the use of the nonlinear projection model, lens distortion is also implicitly rectified.

B. Particle Filter Computation Overview

To achieve faster processing speed, the entire system is implemented in C++, using the OpenCV library. We initialize the model configuration manually using ground truth data. Since we are using the partitioned particle filter, the prediction, likelihood calculation and resampling are performed in each partition for each frame. At the end of the processing of each frame, the estimated posture of the actor is generated by calculating the expectation over all the particles.

For simplicity, we use a zero order model for prediction, which means we predict the state in the next frame by adding a Gaussian noise around the previous state.

The weight calculation is the most important component of the particle filter. Assuming the use of a stereo camera system as input, we develop an explicit depth cue. Without having actual data from a stereo camera in the dataset, we simulate the stereo images by generating virtual depth images offline.

The virtual depth images are generated from the ground truth motion data which is synchronized with the video sequence. The human motion is then projected onto the image plane according to the projection model. The depth map of the actor is formed by the front surface, which is calculated by weighted averaging the depth of the nearest sampled points. Occlusion is also considered by always updating the depth of a pixel with the smallest depth value. Finally, the image is smoothed and Gaussian noise is added to simulate the noise which would occur during actual sensing. The precision of the virtual depth image is about $2cm$, which is comparable to commercially available stereo cameras.

In each time step, the particle weights are calculated from the depth cue, the edge cue, the color cue, and the distance cue, described in Section IV.B below. After the weight calculation, some particles will have large weights while many others will have very small weights, leading to the degeneration problem [21]. Resampling is used to redistribute the particles in the search space while maintaining the PDF, so that more particles are concentrated in the peaks. In the partitioned particle filter, resampling is performed after the likelihood calculation for each partition and after combining the likelihood of all the partitions. The systematic resampling approach is adopted, which is always favorable because of its good performance and ease in implementation [24], [25].

The final weighting function for each particle is obtained by integrating the weights from all the partitions by weighted multiplication. At the end of the processing for each frame, the estimation is generated by calculating the expectation of the configuration distribution.

C. Weight Calculation

The weight for each particle is calculated by comparing the configuration contained in that particle with the image data using different cues. For each cue, we need to extract the corresponding information from the image, and calculate the distance between the predicted configuration and the image. Since in different cues, the scales of the distances are also different, we rescale all the distances into unit scale ranging from 0 to 1 linearly. Then the weight is calculated from the distance using $w = A^{-d'}$. In this equation, A determines the survival rate of the particle filter in this cue [18]. In our experiment, A is set to 100 and the average survival rate is around 0.5.

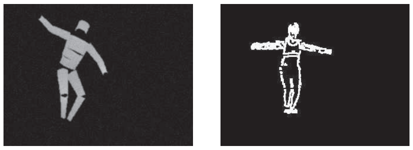
1) Image Pre-processing: First, foreground segmentation is performed, to reduce the effect of background noise. Assuming the background image is available and the camera is static, a standard background subtraction technique [26] is applied on the color image in the RGB space. The subtraction and thresholding are performed separately for R, G and B color. The final foreground mask is obtained by combining the foreground masks obtained from all three colors through Exclusive Or operation. However, because of the complex background, there are many holes on the foreground mask after doing the subtraction. Dilation is performed to fill those holes.

If the camera is moving, for instance in the case of being mounted on a robot head, this background subtraction technique will not be suitable. In this case, if we have good depth image, foreground segmentation can be extracted completely based on depth, and this method is tolerant to ego-motion. Since we are using the virtual depth image, our implementation here is based on background subtraction in the color space.

2) The Depth Cue: Different from the implicit stereo in [15], here we propose an explicit depth cue, in which we use the depth image generated from the stereo system directly. Using the foreground segmentation result, we extract the foreground depth image. The distance for the depth cue is calculated by comparing the projected front surface of the predicted configuration with the foreground depth image. For each sampled point, if its projection is within the foreground mask, the distance is the absolute value of the depth difference. If it is not, the distance is set to a fixed large value as penalty, which is $2.8m$ in our experiment. From our experience, this value should not be too large, otherwise the depth cue will degenerate into a region cue and lose its advantage in providing depth information. The distance for the partition is the summation of the normalized distances for every body part in that partition (Eq. 1). In our current system version, virtual depth images are used instead of the actual depth images (Fig. 2.1).

$$d_{depth} = \sum_{i=0}^{N} \frac{\sum_{j=0}^{M_i} d_{ij}}{M_i} \tag{1}$$

In Eq. 1, d_{ij} is the distance for the jth sampled point of the ith body part in this partition, M_i is the total number of sampled points in body part i, and N is the number of body parts in this partition.



2.1: Depth 2.2: Edge

Fig. 2. Depth Cue and Edge Cue

3) The Edge Cue: To extract the foreground edges, we first apply the Canny operator [27] on the whole grey level image, and then apply the foreground mask to filter out the edges in the background. The remaining edges are strengthened by dilation, as shown in Fig. 2.2. Having the foreground edge image, we go through the projected edges to check whether the projected edges match the extracted edges. The distance is calculated according to Eq. 2.

$$d_{edge} = \sum_{i=0}^{N} \frac{\sum_{j=0}^{M_i}(1 - b_{ij})}{M_i} \tag{2}$$

In Eq. 2, b_{ij} is the binary value for the jth pixel along the predicted edge in the ith body parts, M_i indicates the length of the edge in this body part, and N is the number of body parts in this partition.

4) The Color Cue: The color cue is typically based on template matching, using the color histogram of each body part in the initial frame as the templates. However, we argue that generating the color histogram for each particle takes too much time, because each pixel covered by the human silhouette in the image for each particle must be considered and classified.

Instead, based on the sampled points we already have in the human model, we directly use the colors of these sampled points in the initial frame as the template. The Euclidean distance is calculated from the color of the predicted configuration to the color template in RGB space Eq. 3.

$$d_{color} = \sum_{i=0}^{N} \frac{\sum_{j=0}^{M_i} \sqrt{(r_{ij} - r'_{ij})^2 + (g_{ij} - g'_{ij})^2 + (b_{ij} - b'_{ij})^2}}{M_i} \tag{3}$$

In Eq. 3, r_{ij}, g_{ij} and b_{ij} are the intensity of the red, green and blue color of the jth sampled point in the ith body parts in the predicted configuration, and r'_{ij}, g'_{ij} and b'_{ij} are the values in the template, M_i is the number of sampled points in the ith body part, and N is the number of body parts in this partition.

5) The Distance Cue: During full body tracking, small body parts such as the forearms and the calf legs are the most difficult parts to track. To improve tracking performance, we use an additional cue, the distance cue, which is composed of two parts, the Blob Distance and the Ground Distance. For the Blob Distance, we make use of the two black wrist bands worn by the actor. This does not only help to localize the arms, but also provides a clue for where the torso is. For each arm, the Blob Distance is calculated as the Euclidean distance between the predicted position of the wrist and the corresponding detected band position. For the torso, we consider both wrists at the same time. In addition, we make the assumption that for most of the time, the feet of the actor are always on the ground. The distance is calculated from the feet to the ground. As the Blob Distance, the Ground Distance does not only help to localize the legs, but also helps to localize the torso. For the cases the feet is below the ground, which is obviously impossible, we set a large value for distance as a penalty. For the Ground Distance, we make no specific assumption about the appearance of the feet, so the actor's footwear will not affect the result.

The final weighting function for the lth particle in the ith partition is obtained by integrating these cues using Eq. 4.

$$w_{li} = w_{edge}^{\alpha} \cdot w_{depth}^{\beta} \cdot w_{color}^{\gamma} \cdot w_{blob}^{\delta} \cdot w_{ground}^{\theta} \tag{4}$$

In Eq. 4, w_{li} is the final weight for the lth particle in the ith partition, w_{edge}, w_{depth}, w_{color}, w_{blob}, and w_{ground} are the weights calculated from the distance from the edge cue, the depth cue, the color cue and the distance cue respectively, and α, β, γ, δ and θ which are within $[0, 1]$ are their weights.

IV. Experimental Results

The system is tested using challenging videos, in which the actor performs random and fast motions. The system is shown to be capable of tracking accurately and robustly with high processing speed. Benefiting from the approach we take, the experiment results can be analyzed in both tracking video form and quantitative error form.

A. Tracking Video

Fig. 6 shows the images captured when tracking a video of 1123 frames. In the video, the actor performs a variety of dancing movements, including 11 jumps accompanied by raising arms (Fig. 6.7), half squatting down (Fig. 6.16), and turning (Fig. 6.24). In the experiment shown, 1000 particles are used for the torso, 200 for the head, 500 for each upper arm, 200 for each forearm, 500 for each thigh and 200 for each calf leg. For each cone, the number of sampled points on the front surface is ranging from 15 to 75, proportional to the area of the front surface of the cone.

As can be seen from Fig. 6, the tracker performs well through challenging tracking scenarios, where the actor's movements are random and fast. In the video, the actor moves his body parts through a large range without any constraints. The actor is not restricted to be always directly facing the camera. In the last part of the video, the actor rotates his body about the vertical axis almost 120 degrees. The tracker still keeps tracking, although the accuracy decreases due to the occlusion. The complexity is also reflected in some ambiguous motions for a stereo camera. For example in Fig. 6.2, the forearm is perpendicular to the image plane, and this is difficult to determine from a single view angle. An additional difficulty is the fact that most of the motions are very fast. For example, there are only 17 frames for the jumping motion between Fig. 6.4 to Fig. 6.5, during which the position and the posture of the actor changes significantly. The tracker is able to successfully track through most of these situations smoothly. Even when the tracker loses track of some body parts, it can recover from the incorrect configuration quickly. This can be seen in Fig. 6.9 to Fig. 6.12, where the tracker momentarily loses track of both arms because of the fast movement, but recovers again after 80 frames, demonstrating the robustness of the system.

B. Quantitative Error

In order to perform a quantitative evaluation, an error metric must be defined [8]. Because we are most interested in the final tracking result, our error is measured by calculating the distance from the estimated value to the ground truth. The error is measured in terms of the positions of 15 key joints. Three error terms are defined: ***Average Error*** is calculated by averaging the errors over all the joints in each frame. ***Joint Average Error*** is calculated by averaging the error for each joint throughout the tracking. ***Overall Average Error*** is the average error over all the joints throughout the tracking. In addition to the averages, the standard deviation can also be calculated to measure the fluctuation of the error. In all the following experiments, each test is run for five times.

One of the most basic problems when using a particle filter is how to determine the number of particles. For a partitioned particle filter, the problem is exacerbated because there are several partitions and each can have a different number of particles. Intuitively, the partition with more DOF and closer to the root of the hierarchical structure should have more particles. Empirically, we found that using 1000 particles for the torso, 200 for the head, 500 for each upper arm and 200 for each forearm, 500 for each thigh, 200 for each calf leg, produced good results with an Overall Average Error of $19.1cm$ and standard error deviation of $12.9cm$. These values are used in the following experiments. By using this number of particles, it takes 1.1 second to process each frame. However, the processing speed can be improved by reducing the number of particles. When half the particles are used, it takes 0.67 second for each frame, with an Overall Average Error of $20.7cm$ and standard error deviation of $13.3cm$. If using one fourth of this number, the frame rate can achieve $4Hz$ with an Overall Average Error of $24.3cm$ and standard error deviation of $15.9cm$.

Then, we obtained the Joint Average Error and the corresponding standard deviation, as plotted in Fig. 3. From Fig. 3, we can conclude that the error increases along each kinematic chain. This is because we used more particles for the partitions that are closer to the root, and also the body parts become smaller along the kinematic chain, which makes them difficult to track. Also, the errors for the ankle joints are larger than the wrists. The reason is that the blob distance in the distance cue is more powerful than the ground distance.

Next, the Average Error is generated to help us to determine how the tracker performs as time proceeds. The Average Error is plotted in Fig. 4. From this figure, we can see as time proceeds, the Overall Average Error increases, with several peaks. In the video, the actor performs very small movements in the first 300 frames, followed by 11 sequential jumps which approximately correspond to one peak each in the figure. For example, there is a jump at around frame 400, and another at around frame 590. Starting from frame 850, the actor performs several jumps with body rotation, and the error also increases. From these observations, we can conclude that the error is greater when the actor is performing fast, strenuous, and complex movements.

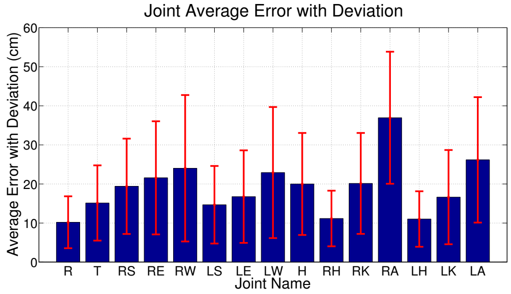


Fig. 3. Joint Average Error with deviation. The joints are the root (R), the thorax (T), the head (H), the two shoulders (RS and LS), the two elbows (RE and LE), the two wrists (RW and LW), the two hips (RH and LH), the two knees (RK and LK), and the two ankles (RA and LA).

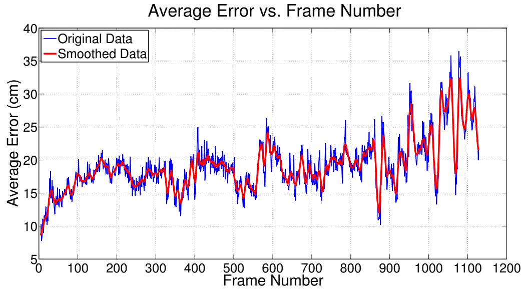


Fig. 4. Average Error

We also tested the tracking performance when using different cues, to determine the relative importance of each cue. In this experiment, we first test the system using all the four cues, and then test excluding one cue to see how this cue affects the tracking. The system is tested using a shorter clip from the video used above, and the result is shown in Table. I.

Surprisingly, the average error is low when the edge cue is excluded. However, we cannot judge the tracking based on the quantitative error alone, but should also consider the correctness from visual inspection. When the edge cue is not used, the two legs usually get overlapped when they are close, and the tracking of the legs is incorrect (Fig. 5), introducing large errors in the leg position. However, tracking of the upper body appears improved because the unusual position of the legs introduces an additional constraint at the waist. This is an artifact of the particular video clip. In order to get the correct tracking result, the edge cue is important to avoid leg overlapping. We can also see that the newly proposed depth cue is the most important cue. Without using the depth cue, the performance decreases significantly, since no other cue incorporates depth information. Adding

TABLE I

CUE COMPARISON (ERRORS REPORTED IN [CM])

	All Cues	No Edge	No Depth	No Color	No Dist.
Error	16.4	14.6	34.8	14.1	19.0
Std Dev	11.7	9.6	23.0	10.7	11.7

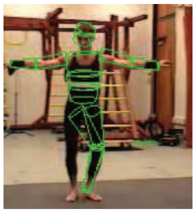


Fig. 5. Tracking without Edge Cue

the color cue also decreases performance, due to the fact that regions in the background are similar to the actor's skin color, and this confuses the tracker.

V. Discussion

Through implementing this tracking system, we investigated several key problems with using a partitioned particle filter for human tracking.

Full body tracking: Through our implementation, we show that by using the partitioned particle filter, full body human motion could be tracked in near real-time. However, partitioning also introduces additional problems, such as how to determine the number of particles and control the survival rate for each partition.

Tracking based on stereo system: Considering the application to portable tracking, the system was designed in a stereo context. Depth information is incorporated into monocular tracking by using the proposed depth cue, and the system achieves successful tracking that would be very difficult when using monocular video only. However, we could benefit further from the stereo system, such as in foreground segmentation. On the other hand, the depth cue could become significantly degraded if the actor is operating in a cluttered environment, where objects in the foreground would affect the depth image.

Cue comparison: We also evaluated the commonly used cues and the newly proposed depth cue. From our experiment, we show that the depth cue is a strong cue which can compensate for the missing depth information in monocular tracking. However, the depth cue cannot work well alone, especially when we only use a limited number of sampled points. The edge cue and distance cue are also strong cues which require less computational resources.

Processing speed: To speed up the system, we applied partitioned particle filter to reduce the required number of particles. For the weight calculation, only sampled points are used to describe the front surface of the human model to reduce computation. To further improve the speed, the most essential problem is how to improve the particle filter to further reduce the required number of particles. According to our measurements, over half of the time is consumed in configuration and projection calculations, so further improvements could be realized by optimizing these computations. In addition, developing more computationally efficient cues would also help to speed up the processing.

Sources of error: For our system, there are several factors that lead to the tracking error. First, the particle filter is based on a sampling method, so the tracking result cannot avoid being jittery when a limited number of particles is used. Especially when the goal is higher processing speed, we have to sacrifice some accuracy. Second, the cues for weighting the particles cannot distinguish certain configurations. For example, when the forearm is perpendicular to the image plane, it is almost impossible for the current system to track correctly. Third, the projection model is not always accurate. In modeling the projection, we made the assumption that the z axis of the world frame is perpendicular to the image plane, although there is a perceptible angle deviation. In estimating the parameters, the sampled points are not uniformly distributed in the work space. So the projection model accuracy also depends on area of the workspace in which the actor is currently located. Finally, the image processing introduces some error. The background is complex and contains regions of similar color with the foreground. When doing background subtraction using color, the foreground is corrupted. This directly affects the edge extraction by losing some edges.

Moreover, we did not consider changes to the environment, for instance lighting brightness. If the brightness changes, the background subtraction and the color cue will be affected, because both of them are done in the RGB space. Also, if a significant part of the human body is not seen by the camera for a long time, the tracking result could be degraded, because the configuration of the unseen body parts will be impossible to estimate and the resulted distribution will be random, and this will affect the configuration estimation of the whole body. If only the upper body is visible, tracking based on a partial model of the body can also be applied [28].

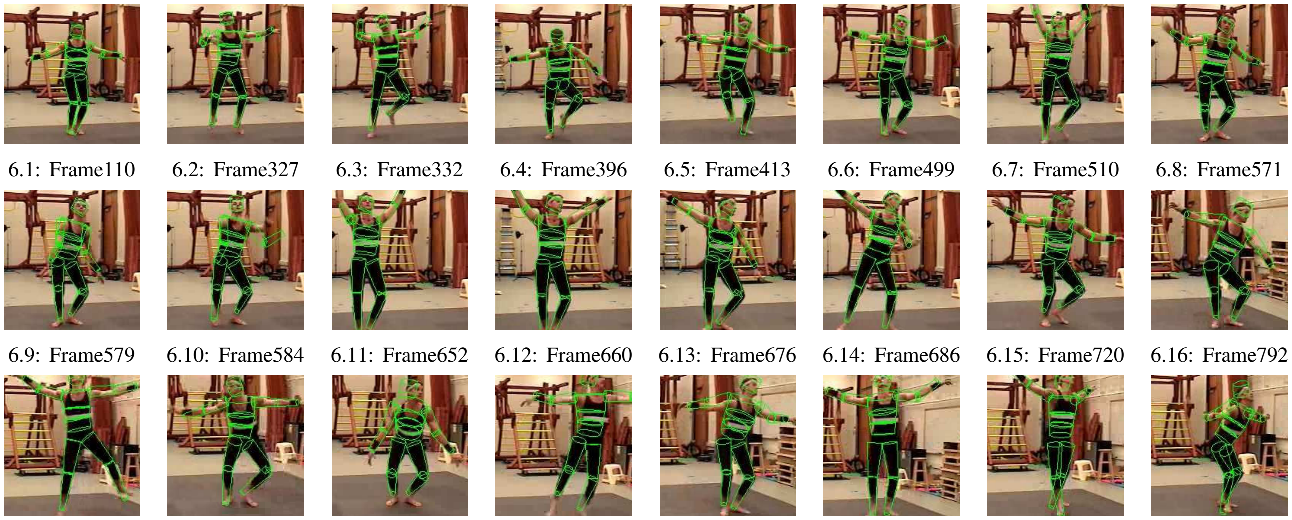
VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a new marker-less full body human motion capture system based on stereo input. The human motion is tracked by a particle filter with partitioned sampling. In addition to the edge, color and distance cue, we propose a new depth cue to utilize the stereo information. To enable a quantitative error analysis of the algorithm performance, we use videos from the publicly available CMU MOCAP database, and generate the virtual depth images offline from the ground truth data. The system is tested with challenging videos, and the results demonstrate our system is capable of tracking random and fast motions accurately and robustly in near real-time.

The current version of the system requires further improvements to realize real-time performance. While tracking performance is excellent when the actor is facing the camera, performance degrades with significant out of plane rotation. Moreover, the standard background subtraction in color space is not suitable for the actual humanoid application, in which case the camera should be able to move. In future work, we hope to address these limitations and implement the system on an on-board camera of a humanoid robot.

References

[1] T. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.



6.1: Frame110 6.2: Frame327 6.3: Frame332 6.4: Frame396 6.5: Frame413 6.6: Frame499 6.7: Frame510 6.8: Frame571

6.9: Frame579 6.10: Frame584 6.11: Frame652 6.12: Frame660 6.13: Frame676 6.14: Frame686 6.15: Frame720 6.16: Frame792

6.17: Frame874 6.18: Frame882 6.19: Frame931 6.20: Frame940 6.21: Frame967 6.22: Frame1022 6.23: Frame1073 6.24: Frame1097

Fig. 6. Frames Extracted from Video

[2] T. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *Computer vision and image understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.

[3] “Vicon.” http://www.vicon.com/.

[4] J. Aggarwal and Q. Cai, “Human motion analysis: A review,” *Computer vision and image understanding*, vol. 73, no. 3, pp. 428–440, 1999.

[5] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard, “Tracking loose-limbed people,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, IEEE Computer Society; 1999, 2004.

[6] J. Deutscher, A. Blake, and I. Reid, “Articulated body motion capture by annealed particle filtering,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, IEEE Computer Society; 1999, 2000.

[7] J. Deutscher, A. Davison, and I. Reid, “Automatic partitioning of high dimensional search spaces associated with articulated body motion capture,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, IEEE Computer Society; 1999, 2001.

[8] A. Balan, L. Sigal, and M. Black, “A Quantitative Evaluation of Video-based 3D Person Tracking,” in *Proceedings of the 14th International Conference on Computer Communications and Networks*, pp. 349–356, IEEE Computer Society, 2005.

[9] J. Bandouch, F. Engstler, and M. Beetz, “Accurate human motion capture using an ergonomics-based anthropometric human model,” *Lecture Notes in Computer Science*, vol. 5098, pp. 248–258, 2008.

[10] H. Sidenbladh, M. Black, and D. Fleet, “Stochastic tracking of 3D human figures using 2D image motion,” *Lecture Notes in Computer Science*, vol. 1843, pp. 702–718, 2000.

[11] P. Azad, A. Ude, R. Dillmann, and G. Cheng, “A full body human motion capture system using particle filtering and on-the-fly edge detection,” in *International Conference on Humanoid Robots (Humanoids), Santa Monica, USA*, 2004.

[12] S. Lin, I. Chang, *et al.*, “3D Human Motion Tracking Using Progressive Particle Filter,” in *Proceedings of the 4th International Symposium on Advances in Visual Computing, Part II*, p. 842, Springer, 2008.

[13] Y. Chen, C. Huang, and L. Fu, “Upper body tracking for human-machine interaction with a moving camera,” in *International Conference on Intelligent Robots and Systems*, 2009.

[14] M. Fontmarty, F. Lerasle, and P. Danes, “Data fusion within a modified annealed particle filter dedicated to human motion capture,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*, pp. 3391–3396, 2007.

[15] P. Azad, A. Ude, T. Asfour, and R. Dillmann, “Stereo-based markerless human motion capture for humanoid robot systems,” in *IEEE International Conference on Robotics and Automation*, pp. 3951–3956, 2007.

[16] M. Sigalas, H. Baltzakis, and P. Trahanias, “Visual tracking of independently moving body and arms,” in *International Conference on Intelligent Robots and Systems*, 2009.

[17] M. Isard and A. Blake, “CONDENSATION - conditional density propagation for visual tracking,” *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[18] J. MacCormick and M. Isard, “Partitioned sampling, articulated objects, and interface-quality hand tracking,” *Lecture Notes in Computer Science*, vol. 1843, pp. 3–19, 2000.

[19] “CMU graphics lab motion capture database.” http://mocap.cs.cmu.edu/.

[20] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[21] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, Feb 2002.

[22] G. Welch and G. Bishop, “An introduction to the kalman filter,” 1995.

[23] J. MacCormick and A. Blake, “A probabilistic exclusion principle for tracking multiple objects,” *International Journal of Computer Vision*, vol. 39, no. 1, pp. 57–71, 2000.

[24] J. Hol, T. Schön, and F. Gustafsson, “On resampling algorithms for particle filters,” in *Nonlinear Statistical Signal Processing Workshop*, pp. 79–82, 2006.

[25] R. Douc, O. Cappé, and E. Moulines, “Comparison of resampling schemes for particle filtering,” in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pp. 64–69, Citeseer, 2005.

[26] M. Piccardi, “Background subtraction techniques: a review,” in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3099–3104, 2004.

[27] J. Canny, “A computational approach to edge detection,” *Readings in Computer Vision: Issues, Problems, Principles and Paradigms*, pp. 184–203.

[28] Z. Li and D. Kulić, “Particle filter based human motion tracking,” in *The Eleventh International Conference on Control, Automation, Robotics and Vision*, 2010.