Particle filter based human motion tracking

(Invited Paper)

Zhenning Li and Dana Kulić
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
z237li@uwaterloo.ca

***Abstract*—This paper proposes a particle filter based marker-less upper body motion capture system, capable of running in realtime. This system is designed for a humanoid robot application, and thus a monocular image sequence is used as input. We first set up a model of the human body, a sub-model which includes 11 Degrees of Freedom is used for the upper body tracking. Considering the realtime processing requirements, two time efficient cues are implemented in the likelihood calculation, namely the edge cue and the distance cue. The system is tested using a publicly available database, which consists of both the videos and the ground truth data, enabling quantitative error analysis. The system successfully tracks the human through arbitrary upper body motion at 20Hz.**

***Index Terms*—realtime, marker-less, human motion capture, particle filter, quantitative error analysis.**

I. Introduction

The goal of computer vision based marker-less human motion capture (MOCAP) is to detect and track human motion through image sequences and to estimate human motion without the use of artificial markers. Human MOCAP has become an active research area, as this technology may lead to significant advances in human activity perception and human computer interaction. Particularly for humanoid robots, using motion data captured in realtime, we are able to develop online autonomous motion learning algorithms, which is an innovative and efficient way to teach a humanoid [1], [2]. Human MOCAP also has many other attractive potential applications in sports training, health rehabilitation, and video surveillance [3].

Several commercial motion capture systems have already been used in the film industry, such as the systems developed by Vicon [4] and Motion Analysis [5]. However, most of these systems are based on markers that are attached on human body. Actors need to wear specific clothing or place markers onto their body, and calibration is needed prior to capturing. In addition, most commercial motion capture systems require a multi-camera (typically 8 to 12) setup, with expensive cameras fixed to permanent locations. These tedious preparations and expensive equipment requirements make current systems unsuitable for widespread use in humanoid robot applications.

In this paper, we propose a realtime marker-less upper body MOCAP system within the particle filtering framework. Designed for a mobile or humanoid robot application, this system uses a monocular image sequence as the input. This reduces computation requirements, but increases the difficulty of the estimation problem, particularly due to missing depth information. We first set up a 3D full body human model, of which a sub-model is used in this upper body tracking. To enable realtime performance, only the edge cue and the distance cue are used in the likelihood calculation. The system has been tested using videos from the Carnegie Mellon University Graphics Lab Motion Capture (CMU MOCAP) Database [6], which includes both the videos and the ground truth data captured using a marker based system. By using this publicly available database, we are able to analyze the system performance quantitatively, which is essential in system evaluation and algorithm comparison. Each frame of the test video is 320 by 240 pixels large, and our system is capable of tracking successfully at a speed of 20Hz on a 2.67GHz Intel Core2 Quad CPU, with an average error of 7cm for each key joint, and an error standard deviation of 6.07cm. Considering the missing depth information, this result demonstrates good accuracy.

The rest of this paper is structured as follows: in Section II, we review the related work on particle filter based human MOCAP and the recent application to humanoid robots. In Section III, the basic principles of the particle filter are introduced. Our implementation is described in Section IV, including the human model, the projection model and the implementation of the particle filter. In Section V, we provide the experimental results and analyze the system performance. Finally, conclusions and directions for future work are given in Section VI.

II. Related Work

Recently, the Bayesian framework has been applied to vision based object tracking and has been proven as an effective method [7]. This approach considers the motion of the tracked object as a state evolution, and solves the tracking problem by estimating the posterior probability density function (PDF) of the state at each time step.

When applying Bayesian filtering to vision based human MOCAP, the system can not be modeled as linear and Gaussian. In general, the Kalman filter fails in this case [8], and the Monte Carlo method [9] is adopted, in an approach named "Particle Filtering" [10]. By using the particle filter, there is no assumption of a linear or Gaussian state distribution, and the PDF can even be multi-modal.

Usually, vision based human MOCAP is performed on an

978-1-4244-7815-6/10/$26.00

articulated human model, whose configuration is determined by the joint angles [11], [12], [13]. Other human models have also been investigated, such as the loosely connected model proposed by Sigal et al [14]. Using the articulated human model, Sidenbladh et al. developed a full body motion tracking system with monocular input [11]. They established the human model consisting of a shape model, an appearance model and a motion model, which transforms the predicted pose state into a predicted image for likelihood calculation. Later, Bandouch et al. developed a motion tracking system based on an accurate anthropometric human model [15].

One of the biggest bottlenecks when applying the particle filter in human MOCAP is the high dimensionality of the human body. The number of particles required for successful tracking increases exponentially with the increase in DOF [16]. For a typical full body human model, there are at least 25 DOF, and this makes the basic particle filter infeasible. Several variants of the particle filter have been proposed to solve this problem, including the partitioned particle filter [16] and the annealed particle filter [17]. Alternatively, this problem can be avoided by setting constraints to reduce the DOF, such as only focusing on the upper body tracking [13], [18], [19].

For most of the existing tracking systems, another limitation is the lack of quantitative error analysis. Without quantitative evaluation, the comparison of different algorithms and the improvement of the tracking performance becomes difficult. In [12], Balan et al. propose the first human MOCAP system with quantitative evaluation. They obtain the ground truth data through a commercial VICON motion capture system and do experiments to compare the performance of a basic particle filter and an annealed particle filter.

Recently, researchers have started to focus on developing marker-less human MOCAP systems applicable for a humanoid robot. Azad et al. develop a system based on stereo input [13], and propose a distance cue, where skin-color hands and head are used as natural markers. Sigalas et al. [18] propose an upper body tracking system also based on stereo input. The pose space is partitioned to cope with the high space dimensionality problem. They combine particle filters with Hidden Markov Models to enable the simultaneous tracking of several hypotheses for the body orientation and the configuration of each of the arms. Yi-Ru Chen et al. [19] propose a partitioned particle filter based upper body tracking system with monocular input. They do not assume a static camera, but rather, the proposed upper body tracking technique adjusts to estimating the human posture during the camera motion. However, the processing speed for the latter two systems are not specified, the test motions for all the three systems are fairly simple, and no quantitative error performance is reported. As a result, it is difficult to evaluate the efficacy of the proposed algorithms.

III. Particle Filter

The particle filter is a method for estimating the evolution of the system state by applying the Monte Carlo method in recursive Bayesian filtering [9]. When applying the Bayesian filter, the system is seen as a dynamic system described by its state, x. When applied to human motion tracking, the state is a vector formed by the joint angles of the human model.

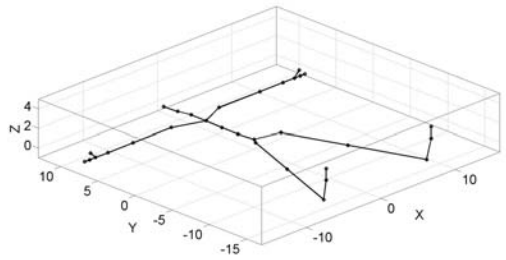


Fig. 1. 3D Full Body Human Skeleton Model

A typical recursive Bayesian filter has two steps, namely a prediction step and an update step. In the prediction step, the prior probability of the state variable $p(x_k|z_{1:k-1})$ at time step k is predicted from the posterior probability at time step $k-1$, $p(x_{k-1}|z_{1:k-1})$ according to the dynamic model $p(x_k|x_{k-1})$ (Eq. 1). In the update step, the prior probability $p(x_k|z_{1:k-1})$ is updated to the posterior probability $p(x_k|z_{1:k})$ by incorporating the observation at time step k, z_k (Eq. 2).

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1}) \cdot p(x_{k-1}|z_{1:k-1}) dx_{k-1} \quad (1)$$

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k) \cdot p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \quad (2)$$

With the assumption of linear system and Gaussian distribution, the optimal estimation is achieved in terms of minimum covariance, i.e., the Kalman Filter [8]. However, the linear and Gaussian assumptions are not usually valid for visual based human MOCAP, in which case the integral is intractable. Using the Monte-Carlo approach, the PDF is sampled and the integral becomes a weighted summation. Each particle contains a hypothesis of the state and a corresponding weight. By storing multiple hypotheses, the particle filter is very robust and can recover from wrong estimations.

IV. System Implementation

A. Human Model

1) Skeleton Model: The motion capture is performed on a 3D articulated human model, whose configuration is determined by the joint angles only. In order to be compatible to the CMU MOCAP database, we construct a highly flexible 3D full body skeleton model (Fig. 1) according to the Acclaim Skeleton File (ASF) convention. Each bone has its own local coordinate frame, and forward kinematics calculation is defined also using the ASF convention.

The full body skeleton model contains at most 30 bones and 64 DOF. For the upper body tracking application, we assume that the actor is always at the same depth and the torso can only rotate in the plane parallel to the image plane. After ignoring the DOF in the back and at the neck, the skeleton model is reduced to 11 DOF, namely 2 DOF for the base translation

TABLE I
BONE DOF

Bone	DOF	Bone	DOF	Bone	DOF
lhipjoint	0	lfemur	3	ltibia	1
lfoot	2	ltoes	1	rhipjoint	0
rfemur	3	rtibia	1	rfoot	2
rtoes	1	lowerback	3	upperback	3
thorax	3	lowerneck	3	upperneck	3
head	3	lclavicle	2	lhumerus	3
lradius	1	lwrist	1	lhand	2
lfingers	1	lthumb	2	rclavicle	2
rhumerus	3	rradius	1	rwrist	1
rhand	2	rfingers	1	rthumb	2

in the plane, 1 DOF for the base rotation, 3 DOF for each shoulder, and 1 DOF for each elbow. This sub-model is used for the upper body tracking.

2) Outer Shape Model: In order to facilitate the contour projection, we designed an outer shape model as a supplement to the skeleton model. Considering the simplicity of the projection, only 2D rectangles are used to represent the body parts that the system is tracking. By using rectangles, we only need to project the four corners and this reduces the computation significantly. For the upper body tracking, we are only interested in the torso, the two arms and the head. As a result, only these body parts have outer shapes.

B. Projection

The posture is described on the human model in 3D, while the observation, which is the image sequence taken by the camera, is in 2D. Therefore, we need a projection model to project the 3D model onto the 2D image plane. The camera is modeled as a pinhole camera for simplicity, and the projection is modeled as weak perspective projection, where the projection matrix is independent of the actual depth of the human subject. The projection is described by a series of coordinate transformations. But because all the transformations are linear, the final transformation can be represented by the projection formula in homogeneous form as shown in Eq. 3.

$$\begin{pmatrix} x_{image} \\ y_{image} \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ b_{11} & b_{12} & b_{13} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{3D} \\ y_{3D} \\ 1 \end{pmatrix} \qquad (3)$$

The projection matrix sufficiently determines the projection. To obtain the projection matrix, 35 equations are found by matching the points in each 2D image and in 3D space, and the projection matrix is estimated through least square surface fitting. The fitting result is listed in Table II and the projection result is illustrated in Fig. 2.

C. Implementation of the Particle Filter

To achieve realtime processing, the entire system is implemented in C++ using the OpenCV library [20]. The particle filter implementation consists of three key components: initialization and state prediction, weight calculation, and resampling and estimation.

TABLE II
FITTING RESULTS FOR THE PROJECTION MODEL

Coefficients (with 95% confidence bounds)	Goodness of fit
$a_{11} = 7.789 \ \ (7.617, 7.96)$ $a_{12} = 0.0955 \ \ (-0.1555, 0.3465)$ $a_{13} = 159.4 \ \ (153.5, 165.3)$	SSE: 327.7 R-square: 0.9954 Adjusted R-square: 0.9951 RMSE: 2.899
$b_{11} = -0.2756 \ \ (-0.5472, -0.003997)$ $b_{12} = -8.802 \ \ (-9.199, -8.405)$ $b_{13} = 336 \ \ (326.7, 345.3)$	SSE: 818.1 R-square: 0.981 Adjusted R-square: 0.9801 RMSE: 4.58

Fig. 2. Projection

1) Initialization and Prediction: Our system is initialized manually using the ground truth data. After initialization, particles are generated by duplicating the initial state. A dynamic model is used to propagate the particles from the previous time step to the current. We use a Zero Order Model as the dynamic model. For each particle, a Gaussian noise is added to the previous state to generate the prediction. We can also consider adding joint limits and limb penetration detection to eliminate impossible postures and reduce the search space [12].

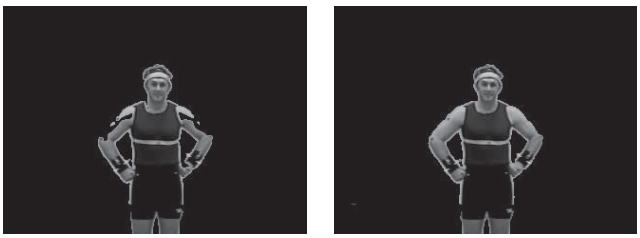
2) Weight Calculation: The core of the particle filter is the calculation of the particle weights, using the cues from the observed image. The cues that can be considered in the human MOCAP include the color cue, edge cue, distance cue, region cue, motion cue and etc. As discussed in [13], the edge cue and the distance cue are the most time efficient cues. Therefore, we implemented these two cues and the final weight is generated from the combination of both cues.

Edges are important sources of information about the shape of the contents of an image. Edge detectors calculate the gradients of intensity in the image, and the edge is obtained by thresholding the gradient image. Canny detector [21] is considered as the best edge detector [22], because it thresholds with hysteresis thresholds and preserves both strong edges and weak edges connected to strong edges. The Canny detector is used in our work.

However, the edges contained in the background make the resulting edge image ambiguous. Background subtraction is applied to segment the foreground before extracting the edges. We tested the segmentation in both the grey image and the color image, and a comparison shows that the background subtraction in the color image is much better (Fig. 3).

After foreground segmentation, the Canny Operator is applied to extract the edges within the foreground region, with dilation applied (Fig. 4(a)).

The distance for the edge cue is calculated by comparing



(a) In Grey Space (b) In RGB Space

Fig. 3. Comparison of Foreground Segmentation in Grey Space and in RGB Space

(a) Dilated Foreground Edges (b) Skin-Color Blobs

Fig. 4. Image Processing Results

the projected edges from each particle with the detected edges in the observed image. For each particle, we go through the projected edges to see whether the corresponding pixels are contained in a detected edge. The distance is calculated according to Eq. 4:

$$d^{edge} = \sum_{i=0}^{n} \frac{\sum_{j=0}^{m_i}(1-b_j)^2}{m_i} \tag{4}$$

where m_i is the number of pixels contained in the edges of the ith body part, and b_j is the binary value for the jth pixel along the projected edge. The distance for each body part is normalized by the length of the edges in that body part, and summed together to form the distance for that particle.

In addition to the edge cue, we also make use of the three skin-color blobs: one face and two hands. Skin color segmentation is done in the YCrCb space, which separates the color information from the brightness information. However, the arms and legs, together with the background also contain skin color. We filter the blobs by area and ratio criteria, and make the assumption that the legs are always lower than the hands, and that the two hands do not cross. The extracted skin-color blobs are shown in Fig. 4(b).

The distance for the distance cue is calculated by summing the Euclidean distances between the predicted positions and the detected positions for all the three blobs. The three skin-color blobs are not always available, but it does not affect the result if any of them is absent for all the particles.

However, the distance calculated from the edge cue and the distance cue are not in the same scale. In order to combine the cues without introducing bias and to increase the resolution, the distances are rescaled into the range of $[0, 1]$ linearly. Then the weight for each cue is calculated from the distance according to the weighting function $w_i = A^{-d_i}$, where w_i is the weight for the ith particle from either edge cue or distance cue, A is a number larger than 1 which affects the sharpness of the weighting function, and d_i is the distance for the ith particle from either edge cue or distance cue.

The final weight for each particle is the normalized combination of the weights calculated from both cues through weighted multiplication.

3) Resampling and Estimation: At each time step, resampling is used to redistribute the particles in the search space while maintaining the PDF, to deal with the degeneration problem [10]. We use the weighting function to resample, which means the number of times each particle is copied is proportional to its weight value. The systematic resampling approach is adopted, which is always favorable because of its good performance and ease in implementation [23]. After resampling, the estimation of the state is computed by calculating the expectation of the posterior PDF.

V. Tracking Results

A. Tracking Video

Fig. 9 shows captured frames from the tracking video when using 300 particles. The system runs at a speed of 20Hz, and the tracking is accurate and robust from visual inspection. The test video contains an actor who performs the ”little tea pot” movement. This movement includes complex motions in every DOF of the model, especially at two shoulder joints. In the video, the green lines indicate the projected skeleton, and the white rectangles indicate the projected outer shape. From the tracking video, we can observe that for most of the time, the system captures the human motion correctly. In Fig. 9(j), however, the tracking result of the head is slightly away from its true position, due to the insufficient DOF in the back and the neck in the skeleton model. Also, the system almost loses tracking for the left arm in Fig. 9(m), but it completely recovers after about 30 frames. This demonstrates the robustness of the particle filter.

B. Quantitative Error Analysis

In order to perform a quantitative evaluation, error metrics must be defined. Because we are most interested in the final tracking result, our error is measured from the expected pose by calculating the distance from the expected pose to the ground truth. The error is measured in terms of the positions of 8 key joints. Three error terms are defined: ***Average Error*** is calculated by averaging the errors over all the joints in one 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 error standard deviation can also be calculated to measure the fluctuation of the error. In all the following experiments, each test is run ten times to compute the error statistics.

We first test the ***Overall Average Error*** when the number of particles increases from 10 to 3000 to find out the optimal number of particles for our system, as plotted in Fig. 5.

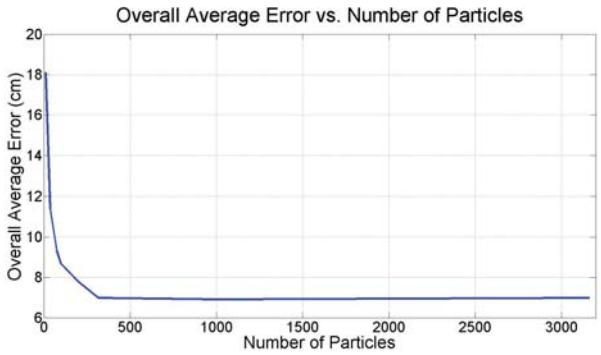


Fig. 5. Overall Average Error

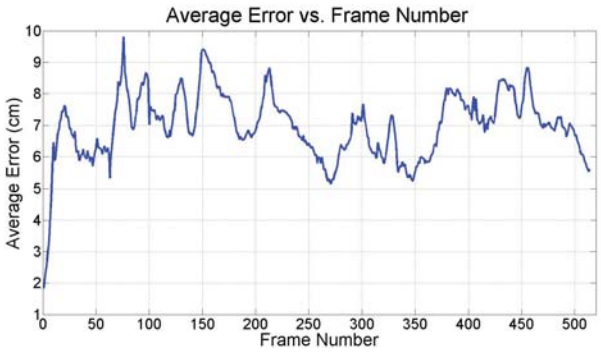


Fig. 6. Average Error Over Frames

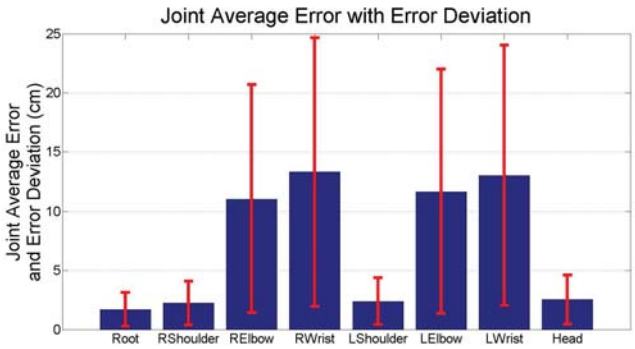


Fig. 7. Joint Average Error with Error Variance

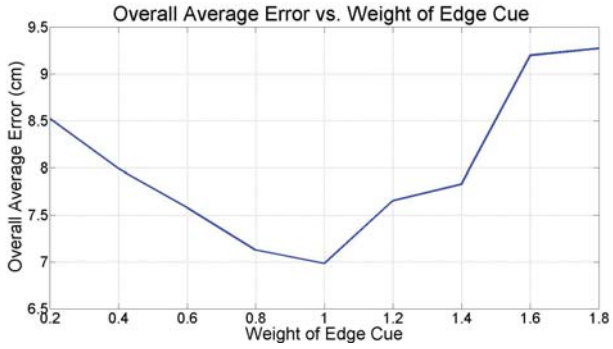


Fig. 8. Overall Average Error When Weights of Cues Change

This figure demonstrates that when the number of particles increases, the error decreases monotonically from 18cm to 6.9cm. Especially, the error drops most significantly from 10 to 300, and then remains almost constant. This indicates 300 particles is the optimal number. When 300 particles are used, the system runs at 20Hz with an overall average error of 7cm and an error standard deviation of 6.07cm.

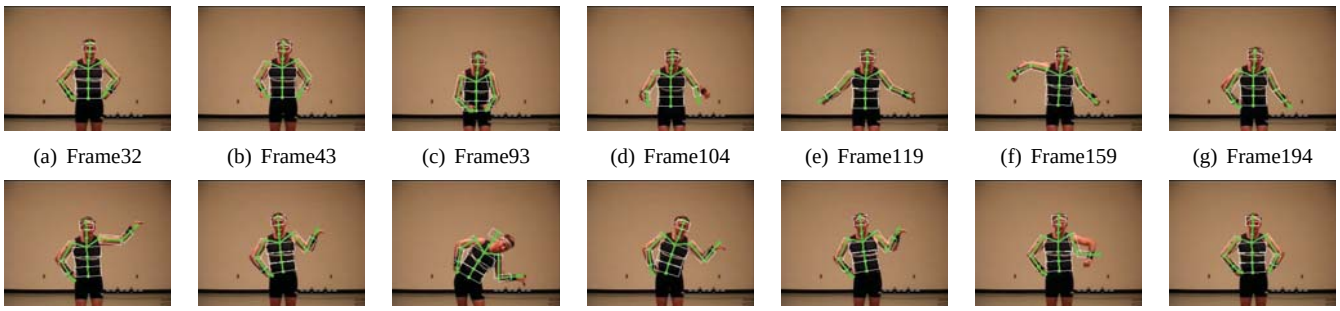
In Fig. 6, we show the ***Average Error*** throughout the tracking when 300 particles are used. The higher errors correspond to the faster motions. Fig. 7 is the bar plot for the ***Joint Average Error*** with error standard deviation. We can see that both the error and the error standard deviation is very small for the root, head and both shoulders. The relatively larger error and error deviation for both arms indicate the arm tracking is poorer than the torso. This is because the arm joints move much faster than the torso within a larger angle range.

The error is mainly due to the missing depth information. In this test video, the base of the actor stays at a certain depth, but the depth of the arms can still change. Without any depth information, the 3D location of the arms becomes difficult to track accurately. The sharp increase of the error from the torso to the arms in Fig. 7 supports this conclusion. Furthermore, the tracker loses track for some body parts occasionally, like the situation in Fig. 9(m). In this case, the tracker loses track for the left arm, while left hand is close to its true position. Here, the edge cue becomes more distinguishing in weighting the particles, and it is more powerful in helping the tracker to recover. If we can adjust the weights adaptively, the system would recover faster and the performance would be improved. Moreover, the limited DOF of the model also introduces error as discussed above.

We also tested the effect of adjusting the weight for each cue. The weight for the edge cue (α) is increased from 0.2 to 1.8 with a step of 0.2, while the sum of the weights remains at 2. This is because after rescaling, the range for the distances calculated from each cue is $[0, 1]$, so the range of the total distance is $[0, 2]$. From Fig. 8, we can see when the weight for each cue is 1, the error is the smallest. This implies both cues are equally important in our system and provide independent sources of information to the tracker. The rescaling for the distances also contributes to this result. Note that the relative cue importance may change, for example if the background also contained many skin-color objects that are difficult to distinguish, the distance cue will become less salient.

VI. Conclusion and Future Work

In this paper, we propose a realtime marker-less upper body MOCAP system within the particle filter framework. Designed for a humanoid application, this system uses monocular image sequences as the input. Successful tracking is achieved through using a particle filter, despite the lack of depth information. To enable realtime human position estimation, the edge cue and the distance cue are used during the likelihood calculation. The system has been tested using videos from the CMU MOCAP database, which includes both the videos and the ground truth motion data captured using VICON system. By using this



(a) Frame32 (b) Frame43 (c) Frame93 (d) Frame104 (e) Frame119 (f) Frame159 (g) Frame194

(h) Frame224 (i) Frame315 (j) Frame416 (k) Frame449 (l) Frame471 (m) Frame491 (n) Frame532

Fig. 9. Frames Extracted from Video

publicly available database, we are able to perform quantitative error analysis which is essential in system evaluation and algorithm comparison.

The current system is based on background subtraction, which limits the application to a static camera. In the future, we plan to remove this assumption, and extend the current system to full body tracking. We also hope to implement the system on a real humanoid using its onboard camera.

ACKNOWLEDGMENT

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

REFERENCES

[1] P. Azad, T. Asfour, and R. Dillmann, "Robust real-time stereo-based markerless human motion capture," in *Proceedings of the IEEE International Conference on Humanoid Robots*, pp. 700–707, 2008.

[2] D. Kulić, D. Lee, Ch. Ott, and Y. Nakamura, "Incremental learning of full body motion primitives for humanoid robots," in *Proceedings of the IEEE International Conference on Humanoid Robots*, pp. 326–332, 2008.

[3] 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.

[4] "Vicon." http://www.vicon.com/.

[5] "Motion analysis." http://www.motionanalysis.com/.

[6] "CMU graphics lab motion capture database." http://mocap.cs.cmu.edu/.

[7] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *Lecture Notes in Computer Science*, pp. 661–675, 2002.

[8] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] G. Bradski and A. Kaehler, *Learning opencv*. O'Reilly Media, Inc., 2008.

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

[22] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "Comparison of edge detectors. A methodology and initial study," *Computer Vision and Image Understanding*, vol. 69, no. 1, pp. 38–54, 1998.

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