PAPER

A Single Camera Motion Capture System for Human-Computer Interaction

Ryuzo OKADA[†a)], ***Member and*** **Björn STENGER**[††b)], ***Nonmember***

SUMMARY This paper presents a method for markerless human motion capture using a single camera. It uses tree-based filtering to efficiently propagate a probability distribution over poses of a 3D body model. The pose vectors and associated shapes are arranged in a tree, which is constructed by hierarchical pairwise clustering, in order to efficiently evaluate the likelihood in each frame. A new likelihood function based on silhouette matching is proposed that improves the pose estimation of thinner body parts, i.e. the limbs. The dynamic model takes self-occlusion into account by increasing the variance of occluded body-parts, thus allowing for recovery when the body part reappears. We present two applications of our method that work in real-time on a Cell Broadband Engine$^{\mathrm{TM}}$: a computer game and a virtual clothing application.

key words: *video motion capture, tree-based filtering, silhouette matching, Cell Broadband Engine^{TM}*

1. Introduction

Human pose estimation from image sequences has various applications in areas such as human-computer interfaces, computer games, and avatar animation, and is an area of active research [2]–[19]. See [20] for a recent review on human body tracking.

Several motion capture systems that use multiple cameras have been proposed, e.g. [6]–[8], [16]. These systems use a shape-from-silhouette approach to estimate the 3D surface and subsequently estimate the 3D pose parameters. Such systems work well under controlled conditions and have been used to capture accurate shape and motion of single actors. In contrast, this paper is motivated by real-time applications such as gesture interfaces. The goal is to build a robust pose estimation system from a single camera. The setup is thus much simpler, however the estimation problem becomes less constrained.

Real-time motion capture has been achieved using incremental tracking, however, in this case the problem of initial pose estimation needs to be solved and often estimation errors can accumulate over long image sequences [12], [19]. Detecting body parts [4], [14], [21], [22] can reduce the computational cost and does not require a manual initial pose estimate, but finding body parts in a single view is particularly difficult because of self-occlusion. Efficient versions of particle filtering have been used with success in the past, but they have the drawback of requiring pose initialization at the start and when tracking failure occurs [5].

Recently learning-based methods have received more attention, where a mapping from observation to body pose is learned from a large set of training examples [2], [23]–[25]. However, these methods do not adapt the final model estimate to an individual subject.

This paper presents a system for real-time pose estimation. Body silhouettes obtained through background subtraction are used as input. We use tree-based filtering for pose estimation, where likelihoods are evaluated in a coarse-to-fine fashion while taking temporal consistency of the poses into account [26].

This paper introduces several innovations that improve the robustness and efficiency:

(1) The 3D body model is selected from a discrete set of models according to the user's body size, is used to generate silhouettes that are used for more accurate matching.

(2) To reduce computation time we evaluate the silhouette distance on an image pyramid using different image resolutions for different tree levels.

(3) The dynamic model explicitly takes self-occlusion into account by increasing the variance of the joint parameters of occluded body-parts. This facilitates tracking these parts when they reappear.

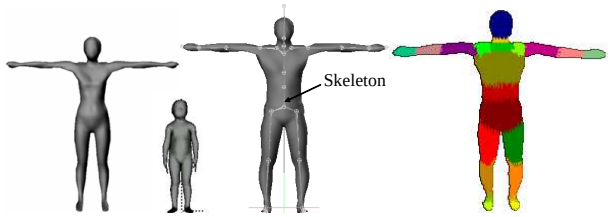
(4) The cost function for silhouette matching is based on weighted distance functions with equal weight on the shape skeleton obtained from the silhouette. Using this normalized weight improves the estimation with respect to thinner body parts such as arms and legs.

2. Tree-based filtering framework

Tracking of body pose is formulated using a probabilistic framework using the standard prediction and update equations: Given the observations up to time t, $\boldsymbol{z}_{1:t}$, the aim is to estimate the posterior distribution of the state $\boldsymbol{x}_t$ which consists of joint angles and 3D global position. The posterior is updated when obtaining the observation at time t:

Manuscript received January 1, 2003.
Manuscript revised January 1, 2003.
Final manuscript received January 1, 2003.

[†]The author is with Corporate R&D Center, Toshiba Corporation
[††]The author is with Toshiba Research Europe Ltd, Computer Vision Group
a) E-mail: ryuzo.okada@toshiba.co.jp
b) E-mail: bjorn.stenger@crl.toshiba.co.uk



(a) Three body models (b) Silhouette

Fig. 1 3D body model. (a) Three body models (out of 30) that are used to represent the variety of body shapes. During the initialization step one of the models is selected. (b) Silhouette of the right body model in (a). The body is represented by a triangular mesh animated by a skeleton with 27 joints. Projected limbs are shown in different colors in the silhouette image.

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = c_t p(\boldsymbol{z}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}), \quad (1)$$

where c_t is a normalization constant and $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ and $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$ are likelihood and prior distribution, respectively. The prior is computed as follows:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \sum_{\boldsymbol{x}_{t-1}} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}), \quad (2)$$

where $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is the distribution for state transitions. The state posterior distribution is estimated in each time step by repeated application of prediction (Equation 2) and update (Equation 1).

We use a point-mass representation for the distributions [27] and approximate the integrals in the filtering equations by replacing continuous integrals with Riemann sums over finite regions. The distributions are modeled as piecewise constant over these regions. In [26] we presented a method that uses a hierarchical partitioning of the state space and represent the distributions by the center points at each partition. In contrast, in this paper we use hierarchical shape clustering to define partitions in state space, i.e. each partition is represented by a cluster prototype and we assume piecewise constant distributions for each of the clusters. Let $\mathcal{R}$ be the compact region of the state space that contains the valid pose vectors. This region is divided into N_l partitions $\{\mathcal{X}^{i,l}\}_{i=1}^{N_l}$ at each tree-level l,

$$\mathcal{R} = \bigcup_{i=1}^{N_l} \mathcal{X}^{i,l} \text{ for } l = 1, \ldots, L\,. \quad (3)$$

Define a discrete probability distribution $p(\hat{\mathbf{x}}_t^{i,l})$ over the regions $\mathcal{X}^{i,l}$,

$$p(\hat{\mathbf{x}}_t^{i,l}|\mathbf{z}_{1:t}) = \int_{\mathbf{x}_t \in \mathcal{X}^{i,l}} p(\mathbf{x}_t|\mathbf{z}_{1:t})\, d\mathbf{x}_t. \quad (4)$$

Given the distribution over the leaves of the tree, $p(\hat{\mathbf{x}}_{t-1}^{i,L}|\mathbf{z}_{1:t-1})$, at the previous time step $t-1$, the prediction equation now becomes:

$$p(\hat{\mathbf{x}}_t^{j,l}|\mathbf{z}_{1:t-1}) = \sum_{i=1}^{N_L} p(\hat{\mathbf{x}}_t^{j,l}|\hat{\mathbf{x}}_{t-1}^{i,L})\; p(\hat{\mathbf{x}}_{t-1}^{i,L}|\mathbf{z}_{1:t-1}). \quad (5)$$

Algorithm 1 Shape hierarchy construction

Input:
Pose data $\{\boldsymbol{x}_j\}_{j=1}^N$
Threshold values of silhouette distance, t_L
Initialization at level $L = 0$
Assign all pose data $\{\boldsymbol{x}_j\}_{j=1}^N$ to a single node.
At level $L > 0$
For each node n on the parent level $L-1$,

1. Select the pose with the smallest mean silhouette distance to the other silhouettes in node n as prototype,
2. Do
   Select the pose with the largest mean silhouette distance d_{max} from the selected poses as prototype
   While $d_{max} > t_L$
3. Cluster by assigning all other poses to their closest prototype.
4. Each cluster forms a node at level L with parent node n.

The transition distributions are estimated by histogramming the transitions in the training data.

2.1 Hierarchy of silhouette shapes

Using a gyro-based motion capture system, pose data is collected from subjects that each perform a variety of motions. Only poses are stored that differ at least 5 degrees in any joint angle from other poses. The pose data is used to generate silhouette shapes by projecting a 3D body model onto the image plane. The geometric model is a triangular mesh whose vertices are attached to a skeleton with 27 joints, see Figure 1. We use a calibrated camera to compute the projection of the mesh onto the image plane.

Hierarchical pair-wise clustering based on silhouette shape distances (see next section) is used to construct the shape hierarchy, see Algorithm 1 and Figure 2. This is similar to the shape hierarchy in [28] but with important differences: (i) the input to the algorithm is the maximum within-cluster distance, and (ii) each node also contains an associated pose vector.

The tree is used for computing a coarse-to-fine approximation of the true posterior distribution over the poses, i.e. the tree is evaluated once for each frame. If a node has a low posterior value in an upper level, the subtree of that node is not searched. The thresholds for this decision are set according to the distance in the clustering step. Note that in contrast to [26], the state space is divided based on the silhouette distance. This avoids the generation of different nodes for cases where poses are nearly identical but one limb is occluded in frontal view.

For further computational efficiency, an image pyramid is used for evaluating the silhouette distance. For a tree of height three image resolutions of 80×60, 160×120, and 320×240 pixels are used at the first to third level, respectively.

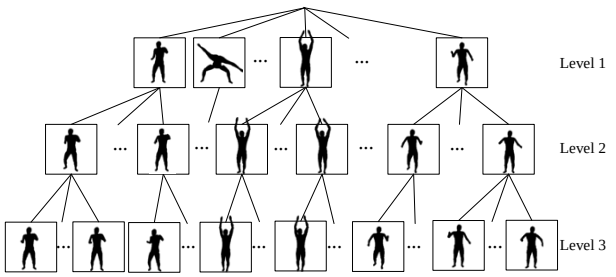
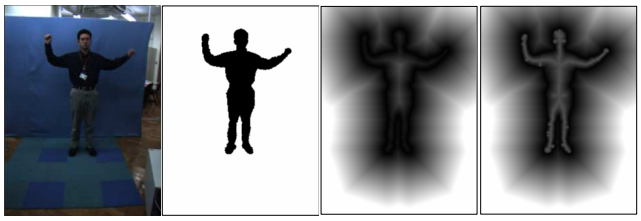


Fig. 2 Shape hierarchy. Each node in the tree contains a pose vector and a silhouette representation. It is generated by hierarchical clustering in the silhouette shapes. The tree contains 57,136 nodes.



(a) Input (b) Silhouette (c) wXOR (d) nwXOR

Fig. 3 Weighted distance functions for likelihood computation: Extracted silhouette (b) from input (a) and the weights for (c) core-weighted XOR and (d) normalized core-weighted XOR, where high brightness corresponds to larger weight. The normalized XOR in (d) makes the pose estimation of limbs more stable.

2.2 Likelihood computation

The likelihood relates the observed silhouette $\boldsymbol{z}_t$ in the current image to the the unknown pose $\boldsymbol{x}_t$. We model the likelihood function with a normal distribution:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t) \sim \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{d(S, T_{\boldsymbol{x}_t})^2}{2\sigma^2}\right), \tag{6}$$

where $d(S, T_{\boldsymbol{x}_t})$ is a silhouette distance between the silhouette S (foreground 1, background 0) observed in the image in the current frame and the model (template) silhouette $T_{\boldsymbol{x}_t}$ generated from the 3D body model in pose $\boldsymbol{x}_t$, and the variance σ^2 is determined experimentally. In the following, we drop the subscript $\boldsymbol{x}_t$ of the model silhouette for simplicity.

The choice of the distance for comparing two silhouettes is crucial, as we require high discriminative power as well as rapid evaluation. A straightforward option is the XOR distance in a fixed bounding window w (which is essentially a Hamming distance):

$$d_{\mathrm{XOR}}(S, T) = \frac{1}{|w|} \sum_{k \in w} (1 - \delta_{S(k),T(k)}), \tag{7}$$

where δ is the Kronecker delta function (1-XOR) and k represents a pixel in the window w. However, since this cost weighs differences close to the contour, equal to those close to the skeleton of the silhouette, it is sensitive to variation of clothing and body shape.

In order to emphasize structural difference between the silhouettes, Chen et al. [29] have suggested a 'core-weighted' XOR distance, defined as:

$$d_{w\mathrm{XOR}}(S, T) = \frac{1}{|w|} \sum_{k \in w} (1 - \delta_{S(k),T(k)}) g(k), \tag{8}$$

where the weight

$$g(k) = D(S)(k) + \alpha D(\bar{S})(k) \tag{9}$$

gives different weight to different types of mismatches. $D(S)$ is the distance transform of the silhouette image S, where pixel values are zero inside the silhouette and increase with the distance from the contour outside the silhouette. $\bar{S}$ is the pixel-wise inverse of the silhouette and its distance transform is zero outside the contour and high in regions near the *core* area, i.e. the shape skeleton. The weight α is set to 5 in [29], thereby highly penalizing the case when points inside the image silhouette are not covered by the model projection.

One drawback of this choice of silhouette distance is that pixels on different parts of the shape skeleton have different penalties. This is because the distance transform $D(\bar{S})$ generally contains higher values for large body parts such as the torso, leading to instability when estimating the pose of thinner limbs. We therefore normalize the weight such that each pixel on the shape skeleton has the same weight. This is done by dividing the right term in Equation 9 by the distance between the contour and the skeleton:

$$g(k) = D(S) + \beta \frac{D(\bar{S})}{D(\bar{S}) + D(S_{skl})}, \tag{10}$$

where S_{skl} and $D(S_{skl})$ are the skeleton of the silhouette shape and its distance transform, respectively. We set the weight β to 1. The weight of a pixel inside the silhouette is normalized by dividing the distance from the silhouette contour, $D(\bar{S})$, by the distance between the contour and the skeleton, $D(S) + D(S_{skl})$. The shape skeleton is defined as the ridge of the values in the distance transformed image $D(\bar{S})$. See Figure 3 for a visualisation of the weights for the normalized core-weighted XOR distance $d_{nw\mathrm{XOR}}(S, T)$. Figure 4 demonstrates the improvement in robustness over other cost functions.

We use integral images for efficiently computing the silhouette distance $d_{nw\mathrm{XOR}}(S, T)$. First we transform the silhouette distance into the following form:

$$d_{nw\mathrm{XOR}}(S, T) = \frac{1}{|w|} \left(\sum_{\{k|T(k)=1\}} D(S) + \sum_{\{k|T(k)=0\}} \frac{\beta}{D(\bar{S}) + D(S_{skl})} D(\bar{S}) \right). \tag{11}$$

Type	Fail	Error	Success
wXOR+ DM1	24	0	0
wXOR+ DM2	24	0	0
nwXOR+ DM1	7	7	10
nwXOR+ DM2	0	10	14

Fig. 4 Tracking an arm swing using the proposed distance function and dynamic model. In 24 experiments, the proposed distance function based on normalized weighted XOR performed most robustly. DM1 and DM2 are two different dynamic models: DM1 does not explicitly handle self-occlusion in contrast to DM2, where Equation 14 is applied. The top shows the three cases corresponding to the table below, namely tracking failure, inaccurate tracking (error) and successful tracking.

We compute a horizontal integral image I of the distance transform $D(S)$, i.e. $I(x,y) = \sum_{j=1}^{x} D(S)(j,y)$, and evaluate the first term in Equation 11 by using the integral values at contour points of the silhouette T:

$$\sum_{\{k|T(k)=1\}} D(S) = \sum_{y=1}^{h} \sum_{j=1}^{n_y} \left(I(x_j^r, y) - I(x_j^l, y) \right), \quad (12)$$

where h is the height of the bounding window w. The contour points are represented by n_y pairs of left x_j^l and right x_j^r horizontal positions of points on the silhouette contour at each vertical position y (see Figure 5). The second term in Equation 11 can be computed in a similar way using a horizontal integral image of $\frac{\tilde{\alpha}}{D(\bar{S})+D(S_{skl})} D(\bar{S})$.

We use color based background subtraction, where we normalize the color values by their intensities and model the pixel-wise distributions with Gaussian pdfs. In each frame, the silhouette is detected as the set of pixel with a Mahalanobis distance larger than a threshold. For changing backgrounds adaptive techniques such as in [22] can be used.

2.3 Motion model

A first order process model is used as a dynamic model, which is easy to evaluate and shows good adaptability to unknown motion:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \sim N(\boldsymbol{x}_{t-1}, \Sigma), \quad (13)$$

where Σ is a diagonal covariance matrix and the variance σ_j for each body part j is determined from the available motion data.

Since we use a single camera, self-occlusion occurs in many cases. In such cases, stable tracking of the occluded parts is difficult because it is not well modeled with the simple dynamic model in Equation 13 during occlusion. However, the system is capable of estimating the occurrence of self-occlusions using the 3D body model by searching for body parts whose projection is not assigned to any pixel. The variances in the dynamic model of the occluded parts is gradually increased:

$$\Sigma = \mathrm{diag}(\sigma_j'^{\,2}),$$
$$\sigma_j' = \begin{cases} \sigma_j & \text{if part } j \text{ is visible} \\ m\sigma_j & \text{if part } j \text{ is occluded} \end{cases}, \quad (14)$$

where $m > 1$ is a parameter for increasing the standard deviation of an occluded part. We use $m = 5$ in our experiments. When the occluded parts become visible again, their parameter estimates have a large variance.

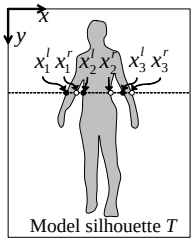


Fig. 5 Silhouette distance computation. The distance in Equation 11 can be computed efficiently by only evaluating horizontal integral images at the silhouette contour, shown here for a single scan line.

3. Experimental results

In the following experiments, we use a color camera with an image resolution of 640×480 pixels, and downsample the image to 320×240 pixels. The pose data is collected by a gyro-based motion capture system using a single subject who performs a variety of motions: turning, side stepping, crouching, Sumo-wrestler's motion (shiko), pointing, kicking, moving both arms upward and downward and golf-swings. The data is captured at 120 Hz and the total number of pose vectors is about 330,000. After eliminating similar poses as described in section 2.1, we have 57,136 poses and construct the tree with three levels as shown in Figure 2, where 7,828, 26,805 and 44,108 nodes are generated for top, middle and bottom level, respectively.

For accurate tracking it is beneficial for the 3D body model to have a similar shape to the current subject. Based on the dress size system in the Japanese Industrial Standard (JIS L4004 and L4005) we obtain 10 body models for men, 14 for women and 6 for children. Before tracking begins we compute silhouette contours and create the tree structure for each body model. All trees and silhouette shapes are loaded into RAM before tracking, requiring approximately 300MB. One of the trees is selected depending to the person to be tracked according to body height, chest size and weight. This selection can be done either from images or, if known, entered by hand.

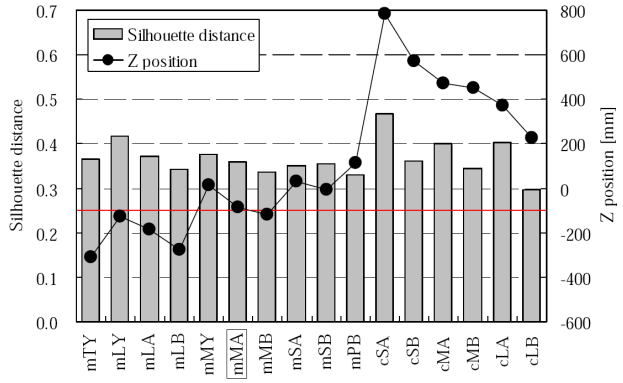
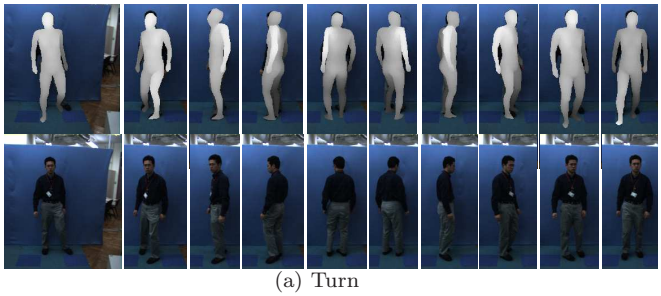
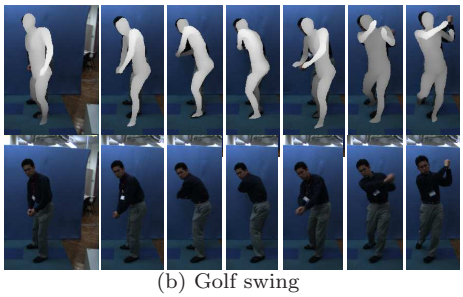


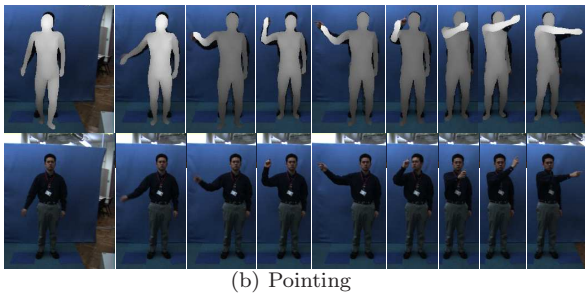
Fig. 6 Silhouette distance and estimated 3D position for different body models. Average measurements on a test sequence with 493 frames. The distance values are similar for various body sizes, but only the model of the measured size 'mMA' and similar models yield measurements close to the ground truth (red line). m and c, represent male and child, respectively. T, L, M, S, and P are indexes of height (T is tallest). Y, A, and B are indexes of chest measurement (B is largest).



(a) Turn



(b) Golf swing



(b) Pointing

Fig. 7 Tracking results during three different actions. The estimated 3D model is superimposed onto each original image in the upper rows, the input images are shown below.

Each new subject is asked to assume an initialization pose at a specific location. The model silhouette is then used to automatically select the best matching body model within the set. For any given input sequence, pose estimation is started when the center of the observed silhouette is close to the image center. Initially we assume that the subject is facing the camera in order to resolve the ambiguity between the frontal and backside pose. In the subsequent frames this ambiguity is handled based on the past pose estimates and the dynamic model.

Figure 6 shows average estimates of global 3D positions in depth (Z-direction) and those of silhouette distances between the observed silhouette and the model silhouette of the estimated pose with respect to body models of 10 males and 6 children. The values are computed on an image sequence with 493 frames of a male subject with measured body size of 'mMA'. The person moves his arms in various ways, crouches and steps sideways. The global position of the body center is approximately at $Z = -100$ mm (see red line). Although the average silhouette distance is similar for the different body shapes, only mMA, mMB, and mLY have average Z values similar to the correct position. The larger sizes mMB and mLY also fit well as all sizes were measured from subjects in underwear and the person in the video is wearing regular clothes.

Figure 7 shows tracking results for three different motions. The frame rate of the input sequence is 15 fps and the duration of the whole sequence is about 2 minutes. In Figure 7(a) the target person turns around. This is a difficult case as it is hard to estimate from a single view which way the person is turning from a side view. In this example the pose is correctly estimated, i.e. the correct mode of the multi-modal pose distribution was chosen here. However, such poses are sometimes estimated incorrectly. In Figure 7(b) the subject performs a golf swing towards the camera. Even though the right arm is occluded, the poses are correctly estimated. Figure 7(c) shows tracking result of the pointing right arm. Our method is capable of tracking such a thin part based on our silhouette distance described in the section 2.2.

One limitation of the system is clearly the dependency on the variability of motion in the database. The method essentially relies on the fact that the database contains a pose close enough to the current input. If this is not the case mis-estimation occurs. Thus the ability to recover is essential. Figure 8 shows some example frames taken from a sequence containing over 3000 frames with a number of different motions. Note that even though pose misalignments occur in some frames, the algorithm is able to recover from these and correctly resume tracking.

The computational time varies depending on the number of likelihood evaluations. The average processing time per frame is 127ms using a high-end PC (two Opteron 280 Dual Core processors with 2.8GHz, 4GB RAM) and 86ms using a Cell Broad-

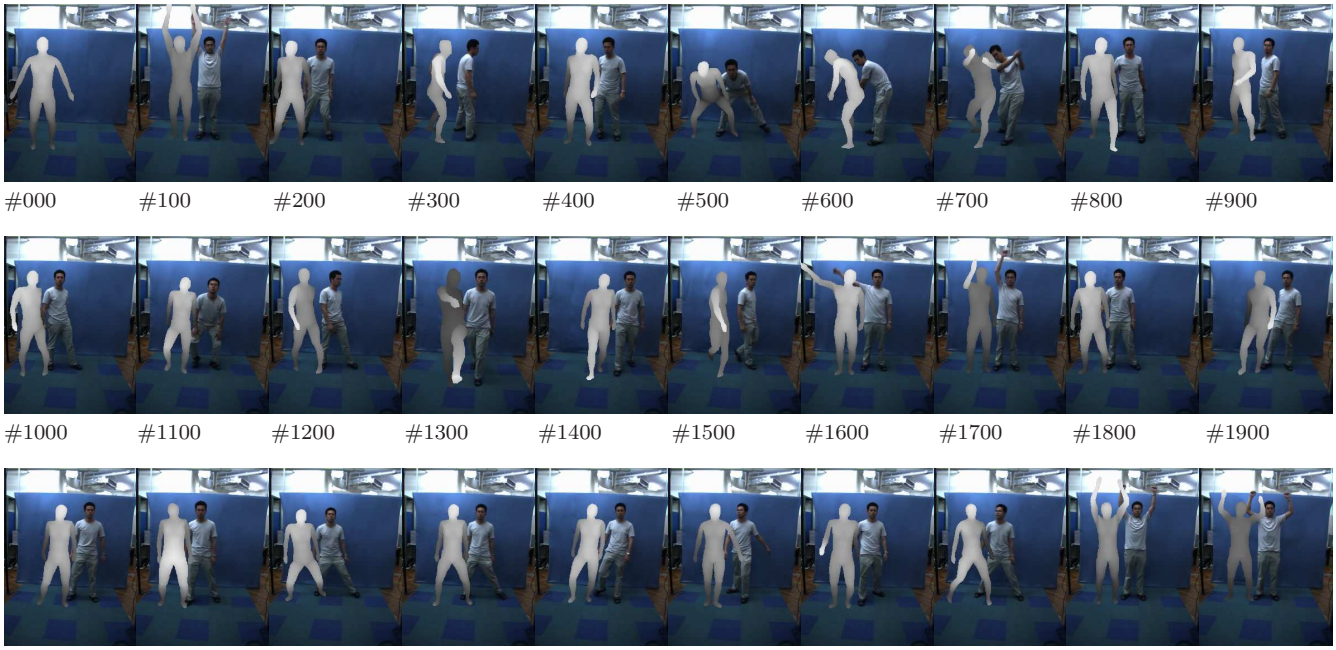


Fig. 8 **Tracking results.** The 3D model in the estimated pose is superimposed onto the input image with an offset. Our system recovers from misestimation occurring in frames #1300 and #1500. In these cases the likelihood values based only on the silhouettes is very similar for multiple poses.

Table 1 **Computation time** in [ms/frame].

	PC			Cell/BE		
	Min	Max	Avg	Min	Max	Avg
Preprocessing	21	29	22	20	53	23
Pose estimation	21	236	104	17	169	62
Total	43	261	127	39	202	86

Table 2 **Number of distance evaluations at each level.**

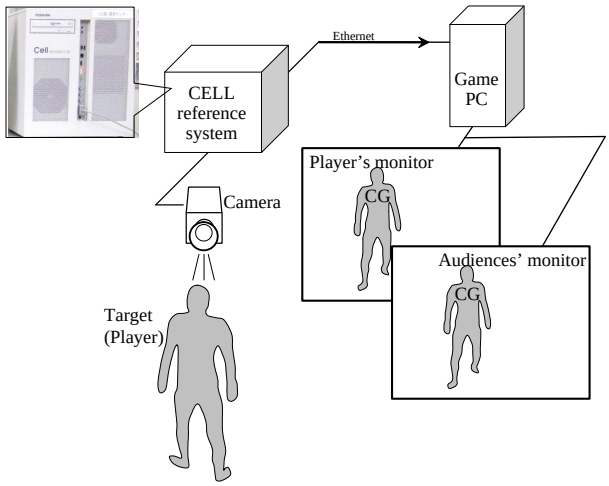
	1st	2nd	3rd
Average	141601.8	43464.5	3497.5
Std. dev.	23147.0	16916.2	385.4

band Engine^{TM}(Cell/BE) [1], respectively (see Table 1). Cell/BE achieves faster processing than the high-end PC by using its parallel computation mechanism of multi-processing cores consisting of a Power Processing Element (PPE) and seven Synergistic Processor Elements (SPEs). In our implementation, parallel processing on SPEs is used for background subtraction on different image partitions, distance transform computation at three image resolutions, and silhouette distance evaluation, computed in parallel on all seven SPEs, respectively. The preprocessing stage includes background subtraction and distance transform with reading a new image frame from the camera. The high maximum value of preprocessing for the Cell/BE system is due to the use of preliminary camera driver software.

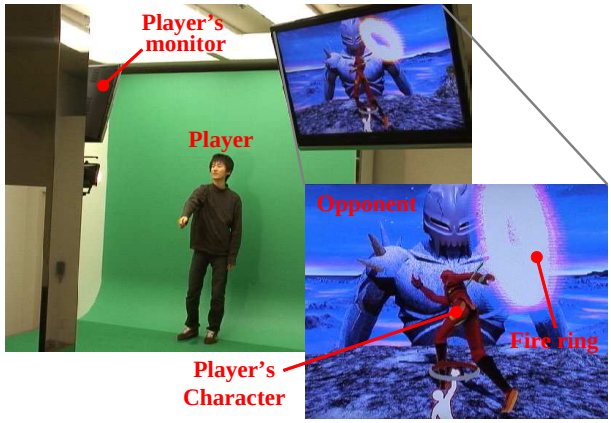
Most of the time for pose estimation is spent on distance computations. When computing the distances according to section 2.2, the processing time is proportional to the image size. Since we use three resolutions of 80×60, 160×120, and 320×240, respectively,the processing time for a single silhouette distance at the first and the second tree level is about 1/4 and 1/2 of that of the third level with full resolution, respectively. Table 2 shows the number of distance evaluations per frame at each level. By using multi-resolution images we obtain an average speed-up factor of 3.1.

4. Application

A computer game based on the proposed method has been developed according to the setup shown in Figure 9. Figure 9(a) shows a system configuration of the computer game. The pose of a player is estimated on a Cell reference set and transfered to a game PC. The game PC uses the pose sequence to move the player's character, a ninja avatar. The player controls the avatar by his/her body motion with the aim to defeat an opponent while avoiding attacks from him. The types of motion that the system recognizes are (1) raising both arms to the sides and bending them on the chest (transfiguration to ninja), (2) crouching (avoid hit 1), (3) stepping sideways (avoid hit 2), (4) swinging the right arm upright and lowering it quickly (hit 1), and (5) raising both arms and lowering them quickly (hit 2). We collected about 400,000 poses at 120 Hz for 3 subjects performing the above motions, and reduced them to about 30,000 poses by eliminating similar poses. The



(a) System configuration of the interactive game



(b) Overview of the interactive game

Fig. 9 Interactive game with visual motion capture. The player's character is attacking the opponent by throwing the fire ring, whose motion is controlled by the player's motion.

system has been demonstrated with approximately fifty different users at an electronics fair.

Additionally, a real-time clothing simulation has been implemented by replacing the game PC in Figure 9(a) with a PC performing avatar control with cloth simulation. As shown in Figure 10, the avatar assumes the same pose as the user and the clothes of the avatar move according to the motion of the avatar using cloth simulation [30]. The pose data and the hierarchy used for obtaining the results in Figure 10 is the same as in the experiments in section 3.

5. Conclusion

In this paper we presented a method for markerless human motion capture using a single camera. It is based on tree-based filtering using a hierarchy of body poses found by clustering the silhouette shapes. We have introduced several improvements such as a person-specific body model, image pyramids for speeding up distance computations, a dynamic model that includes

(a) kick (b) turn

Fig. 10 Clothing simulation with visual motion capture. The avatar moves with the user and its clothes deform according to the avatar's motion.

self-occlusion and a new silhouette distance that improves the estimation with respect to thinner body parts such as arms and legs. A real-time motion capture system was implemented and applied to an interactive computer game.

This work focused on handling automatic initialization and recovery, in order to build a working real-time system. The use of discrete shape templates leads to motion estimates that are less smooth over time than a continuous tracking approach. However, the pose estimates could be smoothed temporally or refined in a continuous optimization step. Temporal integration over a larger time window can help to resolve some cases of ambiguity, e.g. by using the Viterbi algorithm as in [9], [17]. There are many interesting directions for future research, including the combination of our model-based approach with parts-based methods or efficient learning.

Acknowledgements

The authors wish to thank Digital Fashion Ltd. for generating the 3D body models and providing a real-time clothing simulation system, Cavia Inc. for developing the game system, and Kentaro Yokoi of Toshiba Corporate R&D Center as well as FixStars Corporation for implementations on the Cell Broadband Engine™. We are also grateful to Etsuo Fukuda, Nobuhiro Kondoh, and Kazuhiro Hiwada of the Semiconductor Company of Toshiba Corporation for supporting this work.

References

[1] D. Pham and et al., "The design and implementation of a first-generation cell processor," Proc. of IEEE International Solid-State Circuits Symposium, pp.184–186, 2005.

[2] A. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.28, no.1, pp.44–58, 2006.

[3] M. Brand, "Shadow puppetry," Proc. of IEEE International Conference on Computer Vision, pp.1237–1244, 1999.

[4] N. Date, H. Yoshimoto, D. Arita, and R. Taniguchi, "Real-time human motion sensing based on vision-based inverse kinematics for interactive applications," Proc. of IAPR International Conference on Pattern Recognition, pp.318–

321, 2004.

[5] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, pp.1144–1149, 2000.

[6] T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara, "Real-Time 3DShape Reconstruction, Dynamic 3D Mesh Deformation, and High Fidelity Visualization for 3D Video," Int. J. on Comp. Vision and Im. Understanding, Vol.96, No.3, pp.393-434, 2004.

[7] B. Michoud, E. Guillou, H. Briceno, S. Bouakaz, "Real-Time Marker-free Motion Capture from multiple cameras," Proc. 11th Int. Conf. on Computer Vision, 2007.

[8] I. Mikić, M. Trivedi, E. Hunter, P. Cosman, "Human Body Model Acquisition and Tracking using Voxel Data," Int. J. of Comp. Vision, Vol 53, No 3, July/August 2003.

[9] R. Navaratnam, A. Thayananthan, P. Torr, and R. Cipolla, "Hierarchical part-based human body pose estimation," Proc. British Machine Vision Conference, Oxford, UK, pp.479–488, September 2005.

[10] R. Okada, B. Stenger, T. Ike, and N. Kondoh, "Virtual fashion show using marker-less motion capture," Proc. of ACCV, pp.801–810, 2006.

[11] R. Plänkers and P. Fua, "Tracking and modeling people in video sequences," Computer Vision and Image Understanding, vol.81, 2001.

[12] A. Senior, "Real-time articulated human body tracking using silhouette information," Proc. of IEEE Workshop on Visual Surveillance/PETS, pp.30–37, 2003.

[13] H. Sidenbladh, M.J. Black, and D.J. Fleet, "Stochastic tracking of 3D human figures using 2D image motion," Proc. 6th European Conf. on Computer Vision, pp.702–718, 2000.

[14] L. Sigal and M.J. Black, "Predicting 3D people from 2D pictures," Proc. of Conf. Articulated Motion and Deformable Objects, Mallorca, Spain, pp.185–195, 2006.

[15] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," Proc. 9th Int. Conf. on Computer Vision, pp.750–757, 2003.

[16] J. Starck and A. Hilton. "Correspondence labelling for wide-timeframe free-form surface matching," Proc. 11th Int. Conf. on Computer Vision, 2007.

[17] A. Thayananthan, R. Navaratnam, B. Stenger, P.H.S. Torr, and R. Cipolla, "Multi-variate relevance vector machines for tracking," Proc. 9th ECCV, Graz, Austria, May 2006. to appear.

[18] K. Toyama and A. Blake, "Probabilistic tracking with exemplars in a metric space," Int. Journal of Computer Vision, vol.48, no.1, pp.9–19, June 2002.

[19] M. Yamamoto, Y. Ohta, T. Yamagiwa, K. Yagishita, H. Yamanaka, and N. Ohkubo, "Human action tracking guided by key-frames," Proc. of IEEE International Conference on Automatic Face and Gesture Recognition, pp.354–361, 2000.

[20] D.A. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan, "Computational studies of human motion: Part 1, tracking and motion synthesis," Foundations and Trends in Computer Graphics and Vision, vol.1, no.2/3, 2006.

[21] L. Sigal, S. Bhatia, S. Roth, M.J. Black, and M. Isard, "Tracking loose-limbed people," Proc. Conf. Computer Vision and Pattern Recognition, Washington, D.C., pp.421–428, June 2004.

[22] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: Real-time tracking of the human body," IEEE Trans. Pattern Analysis and Machine Intell., vol.19, no.7, pp.780–785, 1997.

[23] V. Athitsos and S. Sclaroff, "Estimating 3D hand pose from a cluttered image," Proc. Conf. Computer Vision and Pattern Recognition, Madison, WI, pp.432–439, June 2003.

[24] R. Rosales, V. Athitsos, L. Sigal, and S. Scarloff, "3D hand pose reconstruction using specialized mappings," Proc. 8th Int. Conf. on Computer Vision, pp.378–385, 2001.

[25] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Learning to reconstruct 3D human motion from bayesian mixtures of experts. a probabilistic discriminative approach," Tech. Rep. CSRG-502, University of Toronto, 2004.

[26] B. Stenger, A. Thayananthan, P.H.S. Torr, , and R. Cipolla, "Model-based hand tracking using a hierarchical bayesian filter," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.28, no.9, pp.1372–1384, 2006.

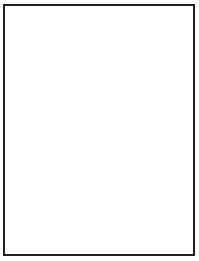
[27] R.S. Bucy and K.D. Senne, "Digital synthesis of nonlinear filters," Automatica, no.7, pp.287–298, 1971.

[28] D.M. Gavrila and V. Philomin, "Real-time object detection for smart vehicles," Proc. 7th Int. Conf. on Computer Vision, Corfu, Greece, pp.87–93, Sept. 1999.

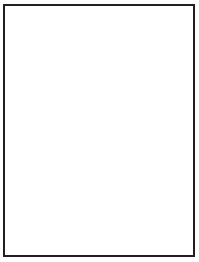
[29] Y. Chen, J. Lee, R. Parent, and R. Machiraju, "Markerless monocular motion capture using image features and physical constraints," Proc. of Computer Graphics International, pp.36–43, 2005.

[30] Y. Sakaguchi, M. Minoh, and K. Ikeda, "Party: A numerical calculation method for a dynamically deformable cloth model," Systems and Computers in Japan, vol.26, no.8-1995, pp.75–87, July 1995.

[31] N. Shimada, K. Kimura, and Y. Shirai, "Real-time 3-D hand posture estimation based on 2-D appearance retrieval using monocular camera," Proc. Int. WS RATFG-RTS, Vancouver, Canada, pp.23–30, July 2001.



Ryuzo Okada received B.E. and M.E. degrees in mechanical engineering from Osaka University in 1995 and 1996, respectively, and finished his studies at the graduate school of the same university in 1999, receiving the D. Eng. degree. He joined Toshiba Corporation in 1999 and was involved in research on computer vision including optical flow estimation, obstacle detection for Intelligent Transport Systems (ITS), and articulated object tracking. He was a visiting researcher at the University of California Los Angeles in 2007.



Björn Stenger received the Diplom-Informatiker degree from the University of Bonn, Germany, in 2000 and the PhD degree from the University of Cambridge, UK, in 2004. He was a Research Fellow at the Toshiba Corporate R&D Center in Kawasaki, Japan, where he worked on gesture interfaces and real-time body tracking. He is currently a member of the Computer Vision Group of Toshiba Research Europe.