A Video Motion Capture System for Interactive Games

Ryuzo Okada
Corporate R&D Center,
Toshiba Corporation
ryuzo.okada@toshiba.co.jp

Nobuhiro Kondoh
Semiconductor Company,
Toshiba Corporation
nobuhiro.kondoh@toshiba.co.jp

Björn Stenger
Toshiba Research Europe Ltd,
Computer Vision Group
bjorn.stenger@crl.toshiba.co.uk

Abstract

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 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. An online motion capture system was implemented on two platforms: a standard PC and a system using Cell Broadband Engine^{TM} [8]. As an application we present a computer game in which an avatar is controlled by the player's body motion.

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 [1, 2, 4, 5, 7, 9, 11, 13, 12, 16, 18].

Some applications, such as gesture interfaces for gaming, require real-time capability, thus an efficient search for the optimal pose is important. 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 [11, 18]. Detecting body parts [4, 13, 17] 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 [1, 10, 14]. However, these methods do not adapt the final model estimate to an individual subject.

In this paper we present a system for real-time pose estimation using a single camera without markers. Our method is based on *tree-based filtering*, where the current pose is estimated by hierarchically evaluating observation likelihoods from image silhouettes while taking temporal consistency of the poses into account [15].

This paper introduces several innovations that improve robustness and efficiency: (1) A 3D body model, selected from a discrete set according to the user's body size, is used to generate silhouettes that are used for more accurate matching. (2) To further increase the computational efficiency, 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 relaxes the temporal constraint on such parts in order to resume tracking them when they reappear. (4) The cost function for silhouette matching is based on weighted distance functions with equal weight on the 'shape skeleton' of 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 pose is formulated using a probabilistic framework as follows: 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 position. With the Markov assumption, that the observation at time t is independent of all past observations given $\boldsymbol{x}_t$, the posterior is updated using the Bayes rule when obtaining the observation at time t:

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

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:

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

where $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is the probability distribution for state transitions. The state posterior distribution is estimated in each time step by repeated application of prediction (Eq. 2) and update (Eq. 1). For computational efficiency a tree structure is used to compute discrete approximations to these distributions.

2.1. Hierarchy of silhouette shapes

Using a marker-based motion capture system, pose data from three subjects is collected. The pose data is used to

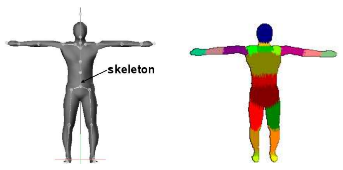


Figure 1. **3D model (left) and its silhouette (right)**. The body is represented by a triangle mesh animated by a skeleton with 27 joints. Projected limbs are shown in different colors in the silhouette image.

generate silhouette shapes by projecting a 3D body model onto the image plane. The geometric model is a triangular mesh and is animated by a skeleton with 27 joints. Hierarchical pair-wise clustering based on silhouette shape distances (see next section) is used to construct the shape hierarchy, see Figure 2. This is similar to the shape hierarchy in [6] 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 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 [15], 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.

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 assume a normal distribution as the likelihood function:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t) \sim \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{d(S_i, S_m(\boldsymbol{x}_t))^2}{2\sigma^2}\right), \qquad (3)$$

where $d(S_i, S_m)$ is a silhouette distance between the silhouette observed in the image S_i in the current frame and the model silhouette S_m generated from the 3D body model in pose $\boldsymbol{x}_t$, and the variance σ^2 is determined experimentally.

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):

$$d_{\mathrm{XOR}}(S_i, S_m) = \frac{1}{|w|} \sum_{k \in w} (1 - \delta_{S_i(k), S_m(k)}), \qquad (4)$$

where δ is the Kronecker delta function (1-XOR). 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.

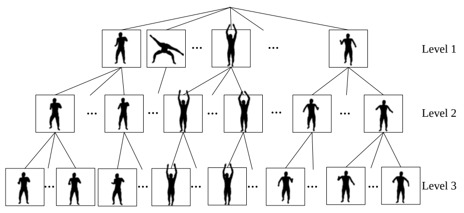


Figure 2. **Shape hierarchy**. Each node in the tree contains a pose vector and a silhouette representation. It is generated by hierarchical clustering of the silhouette shapes. The tree contains 57,136 nodes.

In order to emphasize structural difference between the silhouettes, Chen et al. [3] have suggested a 'core-weighted' XOR distance, defined as:

$$d_{w\mathrm{XOR}}(S_i, S_m) = \frac{1}{|w|} \sum_{k \in w} (1 - \delta_{S_i(k), S_m(k)}) \hat{d}(k), \qquad (5)$$

where the weight

$$\hat{d}(k) = D(S_i)(k) + \alpha D(\bar{S}_i)(k) \qquad (6)$$

gives different weight to different types of mismatches. $D(S_i)$ is the distance transform of the image silhouette S_i (foreground 1, background 0) and is zero inside the silhouette and increases with the distance from the contour. $\bar{S}_i$ is the pixel-wise inverse of the silhouette and its distance function 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 [3], 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 $D(\bar{S}_i)$ 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 Eq. 6 by the distance between the contour and the skeleton:

$$\hat{d}(k) = D(S_i) + \tilde{\alpha}\, \frac{D(\bar{S}_i)}{D(\bar{S}_i) + D(S_{skl})}, \qquad (7)$$

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

We use colour based background subtraction, where we normalize the colour 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 [17] can be used.

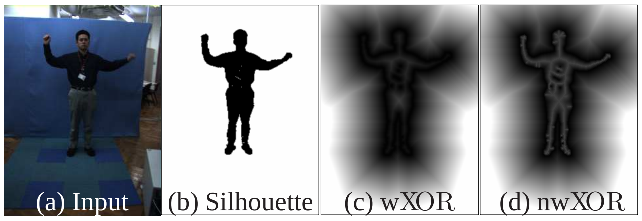


Figure 3. **Weighted distance functions for likelihood computation:** Extracted silhouette (b) from input (a) and the weights for (c) core-weighted XOR and (d) normalized core-weighted XOR, where high brightness corresponds to larger weight. The normalized XOR in (d) makes the pose estimation of limbs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

2.3. 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:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \sim N(\boldsymbol{x}_{t-1}, \Sigma), \tag{8}$$

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 frequently. In such cases, stable tracking of the occluded parts is difficult because the simple dynamic model in Eq. 8 fails 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 variance in the dynamic model of the occluded parts is gradually increased:

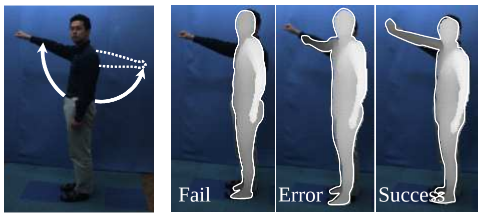
$$\Sigma = \mathrm{diag}({\sigma'_j}^2), \;\; \sigma'_j = \begin{cases} \sigma_j & \text{if part } j \text{ is visible} \\ m\sigma_j & \text{if part } j \text{ is occluded} \end{cases}, \tag{9}$$

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 are observed again, these body parts have large variance and the observation is trusted.

3. Experiment and Application

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 poses are collected by a marker-based motion capture system where the only poses stored are those that differ at least 5 degrees in any joint angle from other poses. The total number of poses is 57,136. We 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 silhouettes and create the tree structure for each body model. All trees and silhouette shapes are loaded into RAM before tracking, requiring approximately 300 MB. One of the trees is selected depending on 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. In our experiments a uniform background is used for stable silhouette extraction, however this is not required as long as the background estimation is sufficiently robust.

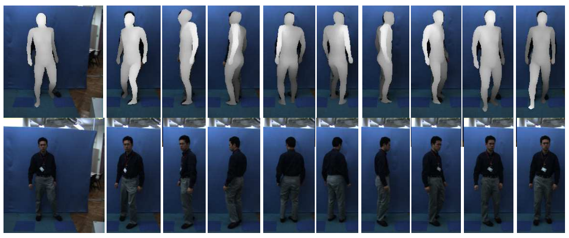


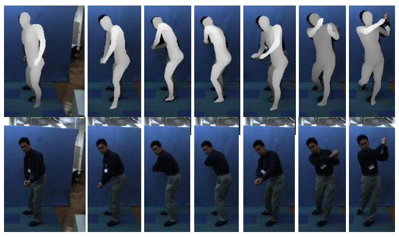
Distance	Fail	Error	Success
wXOR+ DM1	24	0	0
wXOR+ DM2	24	0	0
nwXOR+ DM1	7	7	10
nwXOR+ DM2	0	10	14

Figure 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 9 is applied. The top shows the three cases corresponding to the table below, namely tracking failure, inaccurate tracking (error) and successful tracking.

Figure 5 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 5(a) the target person turns around the axis vertical to the optical axis of the camera, where discrimination between front and back is difficult. The proposed method correctly estimates this motion using the dynamic model. In Figure 5(b) the subject performs a golf swing towards the camera. Even though the right arm is occluded, the poses are correctly estimated. Figure 5(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. The computational time varies depending on the number of likelihood evaluations. The average processing time per frame is 127 ms using a current high-end PC (Two Opteron 280 Dual Core, 2.8 Ghz, 4GB RAM) and 86 ms using a Cell Broadband Engine$^{\text{TM}}$ [8], respectively (see Table 1).

A computer game based on the proposed method has been developed according to the setup shown in Figure 6. A player controls an avatar (“ninja”) by his/her body motion to defeat an opponent or to avoid attacks from the opponent. 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 (hit 1), and (5) raising both arms and lowering them quickly (hit 2). The system has been demonstrated with approximately fifty different users at an electronics fair.


(a) Turn


(b) Golf swing

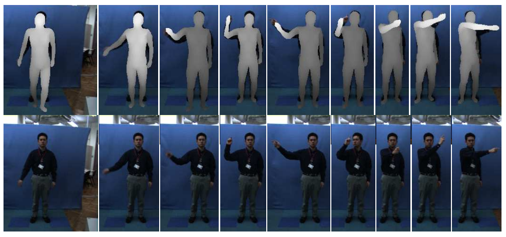

(b) Pointing

Figure 5. **Tracking results.** The estimated 3D model is superimposed onto each original image in the upper rows, the input images are shown below.

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

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

4. Conclusion

We presented a method for markerless human motion capture using a single camera which is based on tree-based filtering using a tree structure of sample poses defined by a silhouette distance. A real-time video motion capture system was implemented and an interactive computer game demonstrated the applicability of the method. The method can also be used for capturing motion of other articulated objects such as hands when a 3D model is available [10, 15]. There are many interesting directions for future research, including the combination of our model-based approach with parts-based methods or efficient learning.

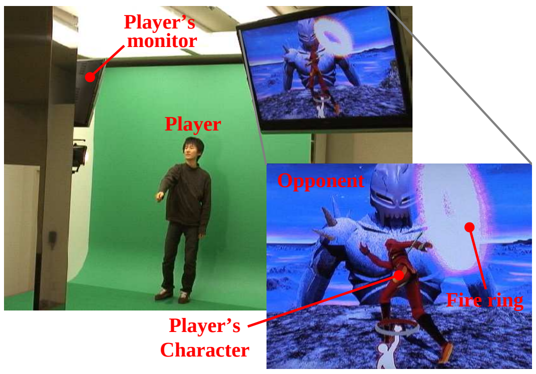


Figure 6. **Interactive game with visual motion capture**. The player's character is attacking the opponent by throwing the fire ring, whose motion is controlled by the player's motion.

References

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *PAMI*, 28(1):44–58, 2006.
[2] M. Brand. Shadow puppetry. In *Proc. of ICCV*, pages 1237–1244, 1999.
[3] Y. Chen, J. Lee, R. Parent, and R. Machiraju. Markerless monocular motion capture using image features and physical constraints. In *Proc. of Computer Graphics International*, pages 36–43, 2005.
[4] N. Date, H. Yoshimoto, D. Arita, and R. Taniguchi. Real-time human motion sensing based on vision-based inverse kinematics for interactive applications. In *Proc. of ICPR*, volume 3, pages 318–321, 2004.
[5] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. of CVPR*, volume 2, pages 1144–1149, 2000.
[6] D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proc. 7th Int. Conf. on Computer Vision*, volume I, pages 87–93, Corfu, Greece, Sept. 1999.
[7] R. Okada, B. Stenger, T. Ike, and N. Kondoh. Virtual fashion show using marker-less motion capture. In *Proc. of ACCV*, pages 801–810, 2006.
[8] D. Pham and et al. The design and implementation of a first-generation cell processor. In *Proc. of IEEE International Solid-State Circuits Symposium*, pages 184–186, 2005.
[9] R. Plänkers and P. Fua. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding*, 81, 2001.
[10] R. Rosales, V. Athitsos, L. Sigal, and S. Scarloff. 3D hand pose reconstruction using specialized mappings. In *Proc. of ICCV*, volume I, pages 378–385, 2001.
[11] A. Senior. Real-time articulated human body tracking using silhouette information. In *Proc. of IEEE Workshop on Visual Surveillance/PETS*, pages 30–37, 2003.
[12] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. 9th Int. Conf. on Computer Vision*, volume II, pages 750–757, 2003.
[13] L. Sigal and M. J. Black. Predicting 3D people from 2D pictures. In *Proc. of Conf. Articulated Motion and Deformable Objects*, pages 185–195, Mallorca, Spain, 2006.
[14] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Learning to reconstruct 3D human motion from bayesian mixtures of experts. a probabilistic discriminative approach. Technical Report CSRG-502, University of Toronto, 2004.
[15] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *PAMI*, 28(9):1372–1384, 2006.
[16] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. Journal of Computer Vision*, 48(1):9–19, June 2002.
[17] 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.*, 19(7):780–785, 1997.
[18] M. Yamamoto, Y. Ohta, T. Yamagiwa, K. Yagishita, H. Yamanaka, and N. Ohkubo. Human action tracking guided by key-frames. In *Proc. of FG*, pages 354–361, 2000.