Virtual Fashion Show Using Real-time Markerless Motion Capture

Ryuzo Okada[1], Björn Stenger[1], Tsukasa Ike[1], and Nobuhiro Kondoh[2]

[1] Corporate Research & Development Center, Toshiba Corporation
[2] Semiconductor Company, Toshiba Corporation
ryuzo.okada@toshiba.co.jp

Abstract. This paper presents a motion capture system using two cameras that is capable of estimating a constrained set of human postures in real time. We first obtain a 3D shape model of a person to be tracked and create a posture dictionary consisting of many posture examples. The posture is estimated by hierarchically matching silhouettes generated by projecting the 3D shape model deformed to have the dictionary poses onto the image plane with the observed silhouette in the current image. Based on this method, we have developed a *virtual fashion show* system that renders a computer graphics-model moving synchronously to a real fashion model, but wearing different clothes.

1 Introduction

In a *virtual fashion show* application the goal is to animate a computer-graphics (CG) model in real-time according to the motion of the real person, while the CG model is wearing a costume different from the actual clothes of the real model. Essentially this task requires an efficient technique for human motion capture with real-time estimation capability.

Currently available commercial motion capture systems require markers or sensors attached to a person. In our system we want to avoid use of visible markers and sensors because fashion models are watched by audiences and we think this is important for variety of motion capture applications in the case of home or office use. One well known approach to vision-based motion capture uses space-carving methods. The shape of a target person is obtained as the intersection of 3D regions generated by inverse projection of silhouettes. This technique [1, 2] requires relatively clean silhouette data obtained from many cameras surrounding the person to be captured. Many approaches that makes use of a 3D shape model of the human body have also been proposed, such as matching feature extracted from captured image and that from the projected 3D shape model [3, 4], learning direct mapping from image features to 3D body pose parameters [5], and defining the force that moves the 3D model to the extracted image feature [6]. These method works with a small number of cameras, but many problems such as stability over long sequences, accuracy, and computational cost remain to be solved. Choosing suitable image feature, such as silhouette [5–7],

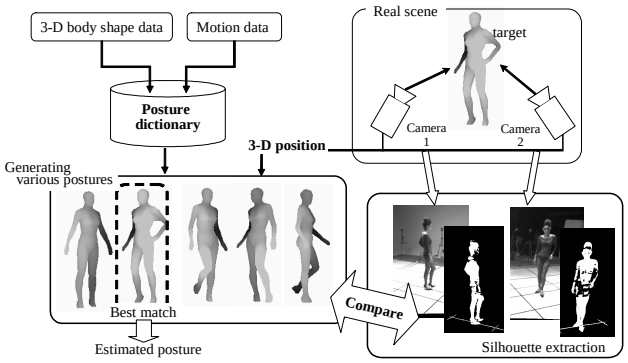


Fig. 1. Overview of the motion capture method

depth [4], and edge [3], depending on an individual target application is one of the important issues. Another problem is how to search for the optimal posture in the high-dimensional parameter space. Real-time motion capture has been achieved using incremental tracking, however, in this case the problem of initial posture estimation needs to be solved [8], and often estimation errors can accumulate over long image sequences [9]. The highly nonlinear relationship between similarity and posture parameters further complicates the problem. In order to address this, versions of particle filtering have been suggested [10, 11], which have been shown to yield good results, given manual initialization and off-line processing. Part-based methods [12] or the use of inverse kinematics [13] may be able to solve the initialization problem and reduce the computational cost of the estimation. However, these methods require the localization of individual body parts, which is difficult in cases where self-occlusion occurs and there are few cameras.

The virtual fashion show application requires real-time processing for synchronizing the motion between the real fashion model and the CG fashion model. Some conditions appropriate for this application can simplify the problem for achieving real-time posture estimation. First, the type of motion is restricted and known beforehand because the motion of the real fashion model is limited to walking and several types of posing. In our setting the fashion model can be required to wear clothes that tightly fit the body, making silhouette matching possible, whose simple definition of cost function also contributes to real-time processing. We are also able to obtain an individual 3D body shape model using a 3D body scanner, as well as posture sequences obtained by a marker-based motion capture system. These data are used to generate a posture dictionary off-line (see section 2 and Fig. 1). Our posture estimation method consists of global position estimation (see section 3) and local pose estimation (see section 4) based on silhouette matching between the observed foreground silhouette and the candidate silhouettes generated from the posture dictionary. We show tracking results and a performance evaluation of posture estimation in section 5 and describe a virtual fashion show in section 6.

2 Posture dictionary

The 3D body shape model is obtained using a laser 3D shape scanner. The number of polygons is reduced from two million to 2000 by deleting vertexes having small curvatures manually in order to achieve a low computational time for silhouette projection. For 640×480 images the time is 1.2–2.0 ms per silhouette projection on a standard PC. The kinematics of the human body are commonly represented by a tree structure whose top node is the body center. Local coordinate systems are defined relative to each body part corresponding to the parent node in the tree structure.

A commercial marker-based motion capture system is used to collect a variety of postures, including walking, posing and turning. A posture captured by the marker-based motion capture system is represented in terms of rotation angles and translation vectors for each joint, which are the parameters to transform a local coordinate system to that of its parent node. Note that the translation parameters are constant except for the body center because the lengths of the body parts do not change, and the parameters of the body center stand for transformation between the local coordinate of the body center and the world coordinate. We call the set of rotation parameters for a posture p a posture vector, which is denoted by $\boldsymbol{r}_p = (r_{p1}, \cdots, r_{p(3N_b)})$, where $N_b = 21$ represents the number of joints.

Due to periodic motion, some poses are very similar, and similar postures are represented by prototype, found by greedy clustering, based on the difference $d_1(a, b)$ between postures a and b:

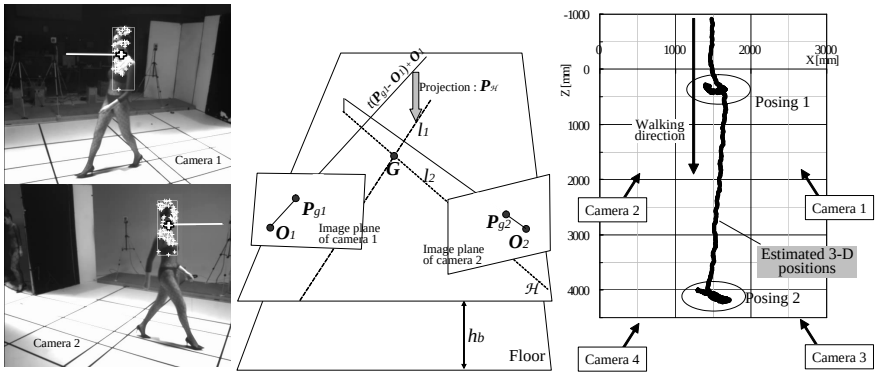
$$d_1(a, b) = \max_{i=1,\cdots,3N_b} |(r_{ai} - r_{bi}) \mod \pi|, \tag{1}$$

which is the largest angle difference of all the rotation parameters. As a result of the clustering, the distances d_1 between any two prototypes are larger than a threshold, which is 7 degrees in our experiments.

3 Global position estimation

For estimating the global body position in the 3D scene, we track the target person in two camera views independently based on our previously proposed tracking algorithm [14]. The algorithm enables us to stably track an object in an image sequence captured at a high frame rate as the motion in the image is very small. In our experiments a frame rate of 100 fps is used. The algorithm consists of corner point tracking and outlier elimination using an affine motion model, and estimates the target position in the image as the mean location of the tracked corner points (see Fig. 2(a)).

Next, we compute the global position of the body center in the world coordinate system by triangulation of the two calibrated cameras using the estimated target positions in the images. The postures that we estimate in the virtual fashion show are all upright, so that the body center moves almost parallel to the



(a) Results of feature point tracking (b) Global position estimation (c) Top view of estimated global positions

Fig. 2. Estimation of global positions of a target person. The tracked feature points are indicated by the white '+' marks on the original images in figure (a). The white rectangle is the tracking window, which is the minimum rectangle containing all feature points. The large white '+' marks are the mean positions of the feature points, which is the estimated target position in the image ($\boldsymbol{P}_{g1}$ and $\boldsymbol{P}_{g2}$), and the white line segment attached to it is the estimated motion vector

floor and the height is approximately fixed to a constant h_b, the height of the body center in standing pose. As shown in Fig. 2(b), we project a line passing through both the camera center $\boldsymbol{O}_c$ and the target position $\boldsymbol{P}_{gc}$ in the image plane onto the plane $\mathcal{H}$, parallel to the floor with distance h_b, and denote the projected line by $\boldsymbol{l}_c$. Assuming that the XZ-plane of the world coordinate system corresponds to the floor, $\boldsymbol{l}_c$ is expressed as follows:

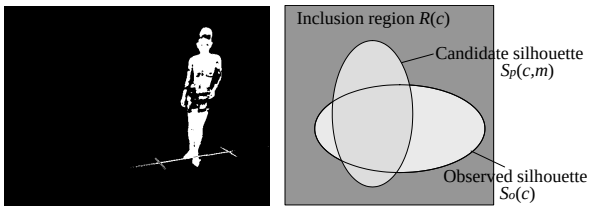
$$\boldsymbol{l}_c = \{\boldsymbol{P}_{\mathcal{H}}(t(\boldsymbol{P}_{gc} - \boldsymbol{O}_c) + \boldsymbol{O}_c) \mid t \in R\}, \quad c \in \{1, 2\}, \quad \boldsymbol{P}_{\mathcal{H}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2)$$

where $\boldsymbol{P}_{\mathcal{H}}$ denotes the projection matrix onto the plane $\mathcal{H}$. The global position $\boldsymbol{G}$ of the target is the point of intersection of the projected lines $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$.

Fig. 2(c) shows results of global position estimation. The target person walks along the Z-axis at $X = 1500$ and poses at $Z = 500$ and $Z = 4000$ shifting the body weight in the X-direction. In this experiment, two pairs of cameras are used to cover the entire area, but one of them is used for global position estimation at each time instance. The area covered by each pair of cameras is determined beforehand and the pair of cameras is selected when the estimated global position is in its predetermined area.

4 Posture estimation

We perform the posture estimation procedure at every fourth frame, i.e. at 25 fps, because the computational cost of the posture estimation is much higher



(a) Extracted Silhouette (b) Similarity

Fig. 3. Silhouette extraction and similarity computation

than that of global position estimation. First, candidate postures that are in the neighborhood of the posture in the previous frame are selected as candidates to restrict the search: We select postures that have similar joint angles to the previous posture p, i.e. the distance $d_1(p, m)$ between p and a selected posture m is smaller than a threshold (60 degrees in our experiments). Since the similarity is defined in terms of silhouette difference in the image (see section 4.1 for details), we impose a further restriction on the number of postures based on an appearance-based distance: We define such an appearance-based posture difference, $d_2(a, b)$, using the positions of joints projected onto the image plane for fast computation as

$$d_2(a, b) = \max_{i=1,\cdots,N_b} |\boldsymbol{p}_{ai} - \boldsymbol{p}_{bi}|, \tag{3}$$

where $\boldsymbol{p}_{ai}$ and $\boldsymbol{p}_{bi}$ denote the positions of joints in the image that are obtained by orthogonal projection of the 3D joint coordinates. We sort the postures selected by $d_1(p, m)$ based on the appearance-based distance $d_2(p, m)$, and select the first n postures as the set M of candidate postures. We use $n = 60$ in our experiments. The silhouettes of each candidate posture m in the set M are generated by (1) translating the 3D body shape model to the estimated global position $\boldsymbol{G}$ in order to correspond the size of the silhouette with that of the observed silhouettes, (2) deforming the 3D body shape model to assume the pose m, and (3) projecting the polygons of the deformed 3D body shape model into each camera view.

The observed silhouette is extracted using background subtraction (see Fig. 3(a)). This often results in noisy silhouettes, but has proved sufficiently stable in our application with reasonably stable lighting conditions.

4.1 Similarity of silhouettes

As shown in Fig. 3(b), $S_p(c, m)$ and $S_o(c)$ denote a candidate silhouette obtained from the candidate posture m, and an observed silhouette for a camera c. $R(c)$ represents the smallest rectangle that contains all candidate silhouettes. The similarity of the silhouettes, $S_p(c, m)$ and $S_o(c)$, should be high when the area of the observed silhouette is large in the candidate silhouette and is small outside the candidate silhouette. Thus, we use the difference between the occupancy rate of the observed silhouette in the candidate silhouette $\rho_i(c, m)$ and that outside

the candidate silhouette $\rho_o(c, m)$ for the similarity normalized with the area of the silhouette:

$$\rho_i(c, m) = \frac{|S_p(c, m) \cap S_o(c)|}{|S_p(c, m)|}, \quad \rho_o(c, m) = \frac{|\overline{S_p(c, m)} \cap R(c) \cap S_o(c)|}{|\overline{S_p(c, m)} \cap R(c)|}, \quad (4)$$

where $|\cdot|$ represents the area of a region.

The similarity measure is affected by the estimation error of the global position. It is therefore necessary to perform optimization for both posture and local shift of the global position. We shift the candidate silhouette in each camera view with a shift $\boldsymbol{d}$, and maximize the similarities independently for each camera in order to optimize the global position locally. Thus, we redefine the similarity for a posture m as

$$s(m) = \sum_c \max_{\boldsymbol{d} \in D} \ (\rho_i(c, m, \boldsymbol{d}) - \rho_o(c, m, \boldsymbol{d})), \quad (5)$$

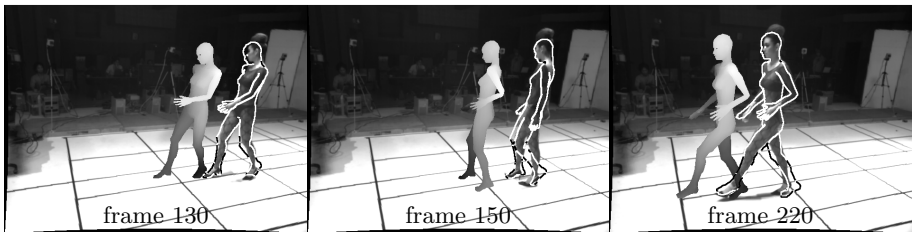
where $\rho_i(c, m, \boldsymbol{d})$ and $\rho_o(c, m, \boldsymbol{d})$ denote the occupancy rate using a candidate silhouette shifted with a shift $\boldsymbol{d}$ in the range of shifts D.

4.2 Hierarchical posture search

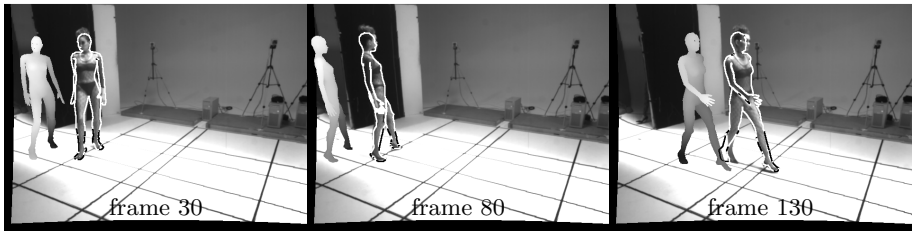
In order to reduce the computational cost of searching for the posture with the greatest similarity, we adopt a coarse-to-fine strategy using a two-level tree, which is generated on-line for each frame. The first layer of the search tree consists of postures selected from M at every t-th posture and the rest of the candidate postures are attached to the closest posture in the first layer as postures in the second level. We search for the optimal posture using the search tree as follows: (1) compute the similarity based on eq. (5) for the postures on the first level of the tree, (2) select the k postures with the greatest similarity, (3) compute the similarity for the postures on the second level in the subtrees of the k selected postures, and (4) select the posture that has the greatest similarity. We use $t = 3$ and $k = 3$ in our experiments.

4.3 Initialization

If a sufficiently large silhouette is extracted in the current image based on the background subtraction, we set the observed silhouette to be the initial target region and start tracking based on our object tracking algorithm [14].When the tracking results come from two or more cameras for the first time, we compute the initial global position, and start the posture estimation with suitable initial posture. Although the initial posture does not fit completely to the posture of the target person, the estimated posture gradually fits to the target person in the subsequent frames.



(a) Results of posture estimation for subject N viewed from camera 3



(b) Results of posture estimation for subject S viewed from camera 2

Fig. 4. Results of posture estimation using four cameras. Estimated postures are indicated by white contour lines on the original image. The 3D figures shown in gray beside the contours represent postures with smoothed motion for CG animation

5 Experiments

Fig. 4 shows the results of posture estimation using four cameras for two subjects who walk and pose differently. In each case the camera arrangement is the same as that shown in Fig. 2(c) and the cameras capture gray-scale images with a resolution of 640x480 pixels at a frame rate of 100 fps. Note that the frame rate for posture estimation is 25 fps as described in section 4. The posture sequence for each subject obtained by the marker-based motion capture system contains about 3600 postures captured at a rate of 120 fps. After clustering the posture vectors the dictionary for each subject consists of about 500 postures, which are experimentally sufficient for our restricted set of motions in the virtual fashion show. Fig. 4 shows that postures are correctly estimated in most frames. In some frames, such as frame 150 in Fig. 4(a), however, the contour lines showing the estimated postures are incorrect.

We have conducted experiments on 27 image sequences for evaluating the performance of the posture estimation. The image sequences include four types of motion shown in Fig. 5(b) performed by three subjects. We use an individual 3D body shape model and motion data for each subject obtained by a laser 3D shape scanner and a commercial marker-based motion capture system, respectively. Table 1 shows the number of misestimations and the number of frames in which misestimation occurs. The number of misestimations, e.g. frame 150 in Fig. 4(a), is counted by comparing the estimate to the ground truth, where postures with small alignment errors are not counted as a misestimation. The misestimation often occurs for a particular subject H compared to the other sub-

Scenario	# Sequence	Frames	Failures	Failure frames	Error in %
S-M1	4	1318	4	12	0.9
S-M2	3	1120	4	87	7.8
S-M3	3	1017	1	17	1.7
S-M4	4	1414	5	37	2.6
H-M2	4	1593	7	127	8.0
H-M3	4	1475	6	64	4.3
N-M1	3	924	6	62	6.7
N-M4	2	694	1	5	0.7

Table 1. Performance evaluation. The first column represents the type of scenario. For example, S-M1 stands for motion sequence M1 performed by subject S. The second column is the number of sequences, which are used in the experiments, and the third column is the total number of frames. Columns four to six show the number of misestimations, the number of frames in which misestimation occurs and the error rate

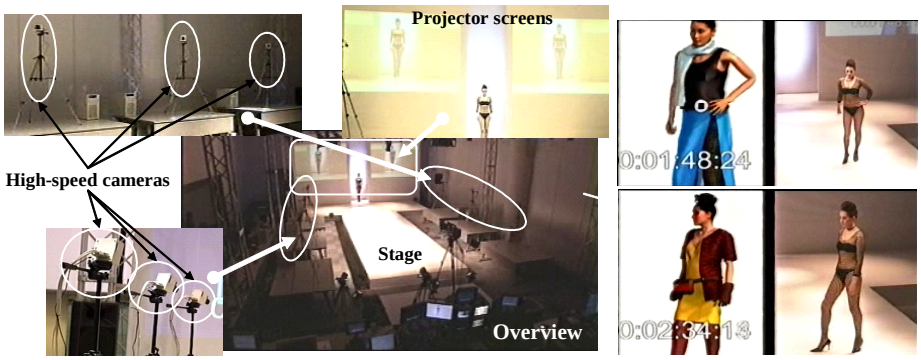
jects. This is because her postures in the image sequences are not contained in the posture dictionary. In total misestimation occurs 34 times for 27 sequences, on average about 1.3 times per sequence (0.089 times per second), corresponding to 4.3 % of the total number of frames. Although we have restricted the search space for posture estimation by selecting candidate postures similar to the previous estimated posture, misestimation occurs for a short period. Such temporal jitter can be reduced by temporal filtering. In our system smooth motion is generated based on the posture sequence recorded by a marker-based motion capture system (see section 6.1). Another reason for the misestimations is the fact that the extracted silhouettes can be very noisy due to shadows on the floor.

6 Virtual fashion show

We have developed a virtual fashion show system using our motion estimation method described in sections 2–4. Fig. 5 shows an overview of the system. A fashion model walks and poses on the stage according to four types of scenarios shown in Fig. 5(b), and our motion capture system estimates her posture. While the fashion model walks along the stage, she poses twice at different positions according to the scenario. Two large projector screens display a full-CG model wearing a costume different from the actual clothes, based on clothes simulation and CG techniques.

6.1 Smooth motion generation

As described in section 5, misestimation of the posture occurs at a certain rate. Even when the posture is correctly estimated, the estimated motion, which is the time series of the estimated postures, is not smooth because the estimated postures can be slightly misaligned. These problems are critical for generating



(a) Virtual fashion show set up (b) Examples of scenarios

Fig. 5. Overview of the virtual fashion show. Three pairs of cameras are placed along both sides of the stage. Two projectors show the CG model wearing clothes different from the actual model as shown in the left images of figure (b)

natural motion of the CG model in the virtual fashion show. We thus combine the estimated motion with the motion data recorded with the marker-based motion capture system.

The recorded motion sequence contains all the postures in the same order as the motion of a real fashion model, except for the timing of walking and posing. We generate smooth motion by changing the playing speed of the recorded posture sequence according to the estimated posture. We start playing the recorded posture sequence when the current estimated posture e is similar to the posture i in the first frame of the recorded posture sequence in terms of the posture difference $d_1(e, i)$.

The 3D figures shown in gray in Fig. 4 represent postures generated by the smooth motion generation method. In the 150th frame in Fig. 4(a) where the posture is misestimated, the motion model finds a plausible posture, even if the silhouette of the estimated posture is slightly misaligned with the observed silhouette. While this smooth motion generation method is straightforward and effective for a specific application of the virtual fashion show, accurate posture estimation and a universal motion generation method are necessary for general applications.

6.2 Hardware configuration

We place three high-speed cameras on each side of the stage (six cameras in total) in order to cover the entire stage which measures about 10 m × 3 m. Each high-speed camera is connected to a PC mounting dual Xeon 3.0 GHz CPUs that captures images and tracks the fashion model in the images as described in section 3. The captured images and the tracking results are transfered to two PCs for posture estimation mounting quad Itanium 2 1.6 GHz CPUs through a high-speed network *Myrinet*, and the two PCs compute the global position and

estimate the posture with smooth motion generation. The estimated posture is sent to a PC for clothes simulation and CG rendering through Gb Ethernet, and the generated CG animation is displayed on two projector screens.

7 Conclusions and future work

We have presented a real-time motion capture system using pairs of cameras and have demonstrated that the system works efficiently for a virtual fashion show based on several constraints appropriate for the virtual fashion show, such as known body shape, tight fitting clothes and limited types of motion.

A possible future application is a virtual try-on for online clothes shopping. However, in order to make this approach work in more general settings, some issues that need to be considered are automatic 3D body shape model acquisition, the use of more robust image features, and efficient matching techniques for increasing the number of postures in the posture dictionary.

References

1. Matsuyama, T., Wu, X., Takai, T., Wada, T.: Real-time dynamic 3D object shape reconstruction and high-fidelity texture mapping for 3D video. IEEE Trans. on Circuits and Systems for Video Technology **14** (2004) 357–369
2. Cheung, G., Baker, S., Kanade, T.: Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In: Proc. of CVPR. Volume 1. (2003) 77–84
3. Gavrila, D., Davis, L.: 3d model-based tracking of humans in action: A multi-view approach. In: Proc. of CVPR. (1996) 73–80
4. Plänkers, R., Fua, P.: Tracking and modeling people in video sequences. Computer Vision and Image Understanding **81** (2001)
5. Agarwa, A., Triggs, B.: 3d human pose from silhouettes by relevance vector regression. In: Proc. of CVPR. Volume 2. (2004) 882–888
6. Delamarre, Q., Faugeras, O.: 3d articulated models and multi-view tracking with silhouettes. In: Proc. of ICCV. Volume 2. (1999) 716–721
7. Brand, M.: Shadow puppetry. In: Proc. of ICCV. (1999) 1237–1244
8. Senior, A.: Real-time articulated human body tracking using silhouette information. In: Proc. of IEEE Workshop on Visual Surveillance/PETS. (2003) 30–37
9. Yamamoto, M., Ohta, Y., Yamagiwa, T., Yagishita, K., Yamanaka, H., Ohkubo, N.: Human action tracking guided by key-frames. In: Proc. of FG. (2000) 354–361
10. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: Proc. of CVPR. Volume 2. (2000) 1144–1149
11. Sminchisescu, C., Triggs, B.: Estimating articulated human motion with covariance scaled sampling. IJRR **22** (2003) 371–391
12. Felzenszwalb, P., Huttenlocher, D.: Efficient matching of pictorial structures. In: Proc. of CVPR. Volume 2. (2000) 66–73
13. Date, N., et al.: Real-time human motion sensing based on vision-based inverse kinematics for interactive applications. In: Proc. of ICPR. Volume 3. (2004) 318–321
14. Okada, R., et al.: High-speed object tracking in ordinary surroundings based on temporally evaluated optical flow. In: Proc. of IROS. (2003) 242–247