2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Model-based motion tracking system using distributed network cameras

Yumi Iwashita, Ryo Kurazume, Takamitsu Mori, Masaki Saito, and Tsutomu Hasegawa

Abstract— **For a coexisting and collaborative society that incorporates humans and robots, the detection, tracking, and recognition of human motion are indispensable techniques for a robot to safely and securely interact with humans. The present paper proposes a motion tracking system using distributed network cameras that are placed in a sizeable environment, such as a street or a town. Model-based motion tracking is adopted in this system, and an asynchronous process is invoked for updating motion estimation in each camera individually. A 2D distance map created by the Fast Marching Method is used to estimate human motion in real-time. Experiments demonstrate that human motion while walking among eight distributed cameras is tracked correctly by automatically selecting appropriate cameras.**

I. INTRODUCTION

As the tasks performed by robots have expanded from the factory assembly line to the everyday human environment, the situation surrounding the use of robots has become increasingly complicated, and the amount of information that must be processed by robots has increased rapidly. However, it is almost impossible for a robot to acquire and process all of this information by means of its on-board computer and sensors, because the capacity and performance of this equipment are quite limited. On the other hand, if the environment surrounding the robot is structured using IT technology such as a distributed sensor network, the robot can perform tasks more reliably and safely, even if the performance of the equipment is limited. This approach involves what is referred to as an Informationally Structured Environment. The basic concept of this approach is that robots provide a variety of services based on environmental information from not only on-board sensors but also sensor networks embedded in the environment.

As an empirical example of the abovementioned approach, we have been involved in the Robot Town Project [1]. The goal of this project is to develop a distributed sensor network system covering a town-size area in which there are several houses, buildings, and streets, and robots manage various services by monitoring the events that occur in the town. The events sensed by the distributed sensors are reported to the Town Management System (TMS), and each robot receives appropriate information concerning its surroundings and instructions for services from TMS. We have already developed a prototype TMS and have demonstrated a number of applications for human-robot collaboration [1] based on several practical scenarios.

Y. Iwashita, R. Kurazume, T. Mori, M. Saito, and T. Hasegawa are with the Graduate Faculty of Information Science and Electrical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka, Japan yumi@ait.kyushu-u.ac.jp

In the present paper, we introduce a human motion tracking system using distributed network cameras in a vast environment, such as a town or a street. This function is one of the fundamental technologies necessary in order for humans and robots to coexist and safely interact. The proposed system enables the detection of human spatiotemporal information, such as position, movement direction, and hand and leg movements.

The remainder of the present paper is organized as follows: Related research is described in Section 2. In Section 3, we introduce the basic algorithm of the proposed tracking system using the 2D distance field and its expansion to the distributed network cameras. Section 4 describes the experimental results of motion tracking for single and multiple individuals using eight distributed network cameras. Conclusions are presented in Section 5.

II. RELATED RESEARCH

Image-based motion tracking using a single camera or a few cameras can be divided into two categories, i.e., learning-based motion tracking and model-based motion tracking. Learning-based motion tracking [2]–[5] is a method by which several pairs of human postures and image features are collected and stored in a database during a learning phase. During the execution phase, human postures are estimated by interpolating postures that fit the acquired image features in the database.

Model-based motion tracking, on the other hand, directly compares a 3D model of a human body and acquired camera images and subsequently estimates the appropriate posture of the 3D model [6]–[18]. A standard processing flow of this technique is as follows: i) a 3D model of a human body with an arbitrary initial posture is projected on a camera image, ii) image features of the projected and acquired images, such as edges and contours, are compared, and iii) the optimum posture in which these features coincide with each other is selected as the estimated posture. However, for the case in which the calculation cost of the image features is large or several cameras are used simultaneously, an effective tracking algorithm is essential when considering real-time processing. In particular, since a time delay induces a critical problem for human-robot interaction in terms of feeling and safety, the latency of the motion estimation should be reduced as much as possible.

In the present study, we develop a real-time motion tracking system using distributed network cameras. Key techniques of this system are the use of the 2D distance field in a camera image [19], which is constructed quite rapidly using the Fast Marching Method [20], and the

asynchronous procedure for model-based motion estimation, which is suitable for distributed network cameras.

Sminchisescu et al. proposed a technique using the 2D distance field based on silhouette images [21], that is similar to the technique proposed herein. However, the proposed technique allows fast construction of the 2D distance field in a coarse-to-fine manner and real-time integration of motion estimation using distributed network cameras.

III. Motion tracking system using the 2D distance field

A. 2D-3D alignment of a rigid object

The present study uses a fast tracking technique for a rigid object using a 2D distance field constructed in a coarse-to-fine manner [19], [22]–[24]. The 2D distance field is a map that shows the minimum distance from a point or a line to another point on the image plane. This technique was developed for the fast 2D-3D alignment of a rigid object and can be performed faster than the conventional Iterative Closest Point (ICP) method because there is no need to search point correspondences, which is a costly procedure in the conventional ICP method.

A brief description of the 2D-3D alignment procedure for a rigid object is as follows:

1) Prepare or capture an object image and a 3D model of the object. We assume that this model consists of a number of small triangle patches.
2) Extract a contour line of a silhouette of the object in the 2D image using an Active Contour Model such as Snakes or the Level Set Method [20], [25].
3) Construct a 2D distance field using Fast Marching Method [20] from the contour line of the silhouette using the coarse-to-fine approach explained herein later.
4) Place a 3D model of the object at an arbitrary position and posture in 3D space.
5) Project the 3D model of the object on the 2D image and find the triangular patches on the occluding boundary of the projected image.
6) Read the 2D distance value on the occluding boundary of the projected image extracted in Step 4. This value indicates the alignment error between the 2D image and the 3D model. Therefore, the total alignment error is calculated by determining the sum of the 2D distance values on the entire occluding boundary. Gradient vectors of the alignment errors can be calculated at the same time.
7) Calculate the compensation values of the position and posture of the 3D model in order to reduce the alignment error using gradient vectors.
8) Repeat Steps 1 through 6 until the 2D image and the projected image of the 3D model coincide with each other.

We expand the above technique for motion tracking of a human body including some rigid bodies and joints using distributed network cameras.

B. 3D Human Body Model

The developed system uses the 3D human body model shown in Fig. 1 [1]. This model consists of 14 links and 13 joints. The numbers assigned to the coordinate systems in Fig.1 indicate the degrees-of-freedom of the joints. The total number of degrees-of-freedom of the model is 21.

Note that we assume the position and posture of the 3D model to be defined by the 3D position of the body center, the rotation angle around the vertical line passing through the center of the body, and 16 joint angles of eight joints, not including the neck, wrists, and ankles. The 3D model is composed of a number of small triangular patches of similar size.

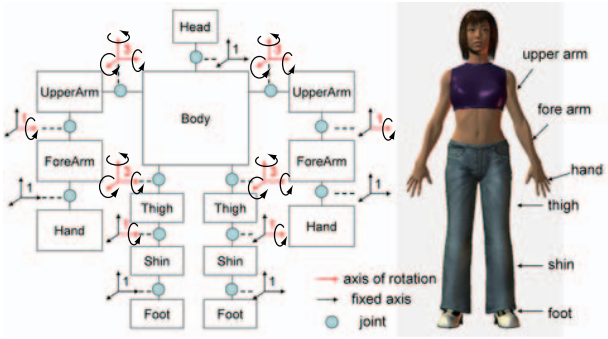


Fig. 1. 3D human body model[1]

C. Construction of the 2D distance field and posture estimation

1) Construction of the 2D distance field: The 2D distance field T in this system is a map that indicates the minimum distance from a point on the contour line I_h of the body silhouette to an arbitrary point on the image. Therefore, T on the point $p = (x_p, y_p)$ satisfies the following constraint:

$$T(p) = 0, p \in I_h \tag{1}$$

$$\mid \bigtriangledown T(p) \mid = 1 \tag{2}$$

To create the 2D distance field, a camera image is captured and a silhouette image of a human is extracted by first subtracting the background image. Next, the Level Set Method is applied to obtain the contour line of the human body. The 2D distance map T is then constructed using the Fast Marching Method. In this step, we adopt a coarse-to-fine approach, that is, a dense 2D distance field is constructed around the contour line using a high-resolution image. On the other hand, a coarse 2D distance field is created for the regions that are far from the contour line by decreasing the resolution of the image. An example of the 2D distance field is shown in Fig.2.

[1]The original model was obtained from Cyberware, Inc. http://www.cyberware.com/products/scanners/wbxSamples.html

Original image Contour image 2D distance field

Fig. 2. Construction of the 2D distance field

2) Open Dynamics Engine, ODE: To imitate a natural motion of a human, dynamics parameters such as the mass and the inertial force should be considered. For example, when a human takes a step forward, the knee joint becomes free temporarily, and a natural walking motion is realized as a step forward by the inertial force under the knee joint.

To consider these dynamic effects, we adopt the Open Dynamics Engine (ODE) for the calculation of whole-body motion. The ODE is a free dynamic simulator developed by Dr. Russell Smith [26], and the dynamic simulation of multiple body dynamics can be performed in real time.

3) Calculation of compensation values: Compensation values to reduce the residual error are calculated as follows: First, the 3D model of a human body is projected on the 2D distance field, T. Here, a point on the contour line of the projected 3D model is $p_i = (x_{p_i}, y_{p_i})$, and the corresponding triangle patch is u_i. The force f_i applied to the patch u_i parallel to the 2D image plane is defined as follows:

$$f_{u_i} = \begin{bmatrix} T(p_i)T_x(p_i)/D(p_i) \\ T(p_i)T_y(p_i)/D(p_i) \end{bmatrix} \tag{3}$$

$$D(p_i) = \sqrt{T_x(p_i)^2 + T_y(p_i)^2} \tag{4}$$

where T_x and T_y are the derivatives of T in the x and y directions, respectively. The force F applied to the center of the 3D model of the human body is obtained as follows:

$$F = \sum_{u_i} f_{u_i} \tag{5}$$

By applying the force F to the center of the 3D model, the position is updated.

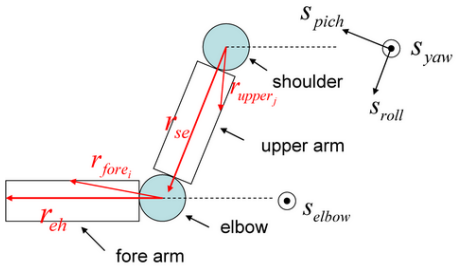


Fig. 3. Arm model

Next, the compensation values for the joints are calculated. For simplicity, we first consider the arm part in Fig.3. Here, we denote a patch corresponding to the projected contour line of the forearm as u_{fore_i}, the vector from the elbow to the patch u_{fore_i} as r_{fore_i}, and the force applied to the patch u_{fore_i}, which is parallel to the image plane, as f_{fore_i}. The moment around the elbow joint is calculated as follows:

$$M_{elbow} = \sum_{u_{fore_i}} (r_{fore_i} \times f_{fore_i}) \tag{6}$$

Therefore, the compensation torque of the elbow joint is calculated as

$$\tau_{fore} = s_{elbow} \cdot M_{elbow} \tag{7}$$

where s_{elbow} is the axis of rotation of the elbow joint.

Next, we denote the patch corresponding to the projected contour lines of the upper arm as u_{upper_j}, the vector from the shoulder to patch u_{upper_j} as r_{upper_j}, the force applied to patch u_{upper_j} parallel to the image plane as f_{upper_j}, and the vector from the shoulder to the elbow as r_{se}. The moment around the shoulder joint is calculated as follows:

$$\begin{aligned} M_{shoulder} &= \sum_{u_{upper_j}} (r_{upper_j} \times f_{upper_j}) \\ &+ \sum_{u_{fore_i}} ((r_{fore_i} + r_{se}) \times f_{fore_i}) \end{aligned} \tag{8}$$

Therefore, the compensation torques of the shoulder joint are calculated as

$$\tau_{roll} = s_{roll} \cdot M_{shoulder} \tag{9}$$

$$\tau_{pitch} = s_{pitch} \cdot M_{shoulder} \tag{10}$$

$$\tau_{yaw} = s_{yaw} \cdot M_{shoulder} \tag{11}$$

where s_{roll}, s_{pitch}, and s_{yaw} are the axes of rotation of the shoulder joint.

From these compensation torques, the compensation angle of each joint is calculated by the ODE by taking the inertia and mass of the body, arms, and legs into consideration.

frame 0 frame 1 frame 2

Fig. 4. Examples of motion estimation

4) Implementation of the Level Set Method on GPU by CUDA: The Level Set Method [25], which extracts the silhouette contour, can be implemented in a parallel manner because this procedure is executed individually at each pixel. Therefore, we use the CUDA programming language, which enables the execution of various scientific calculations on a Graphics Processing Unit (GPU) and the implementation of the Level Set Method on a GPU. Using the GPU, the calculation time of the Level Set Method for an image with 640 x 480 pixels is 14.7 [ms], and real-time processing is realized.

D. Estimation of hand and foot positions by skin color

For the case in which the legs and arms overlap other body parts in the camera image, it is impossible to extract exact contour lines of the legs and arms, and so the correct compensation values cannot be obtained. Therefore, we use

the information of the skin color of the hands and feet in addition to the alignment technique explained above.

First, we project the 3D model on the camera image and calculate vectors f_{hand} from the projected hand and foot positions of the 3D model to the hand and foot positions extracted by skin color. Next, this vector is back projected to the 3D coordinate, and the 3D vector f'_{hand} is calculated. The new compensation torque of the elbow joint is obtained as

$$M'_{elbow} = r_{eh} \times f'_{hand} \tag{12}$$

$$\tau'_{elbow} = s_{elbow} \cdot M'_{elbow} \tag{13}$$

where r_{eh} is the 3D vector from the elbow joint to the hand.

In the same manner, the compensation torque of the shoulder joint is obtained as follows:

$$M'_{shoulder} = (r_{se} + r_{eh}) \times f'_{hand} \tag{14}$$

$$\tau'_{roll} = s_{roll} \cdot M'_{shoulder} \tag{15}$$

$$\tau'_{pitch} = s_{pitch} \cdot M'_{shoulder} \tag{16}$$

$$\tau'_{yaw} = s_{yaw} \cdot M'_{shoulder} \tag{17}$$

All the compensation torques for the elbow and shoulder joints of both arms are calculated using Eqs. (7), (9) (11), (13), and (15) $\sim$ (17), the compensation angle for each joint is calculated using the ODE. This procedure is also performed for both legs. Figure 4 shows an example of alignment using the 2D distance field and the skin color.

E. Motion tracking system using distributed network cameras

Since the procedure explained above is for a single camera, we expand this system to a distributed network camera system. The system configuration is shown in Fig. 5.

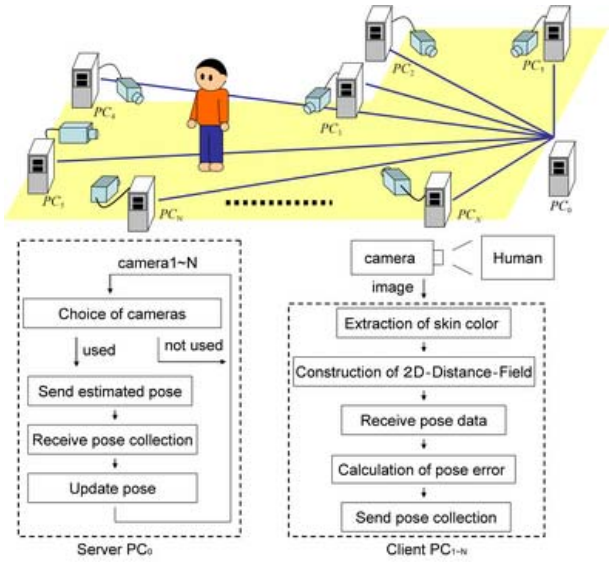
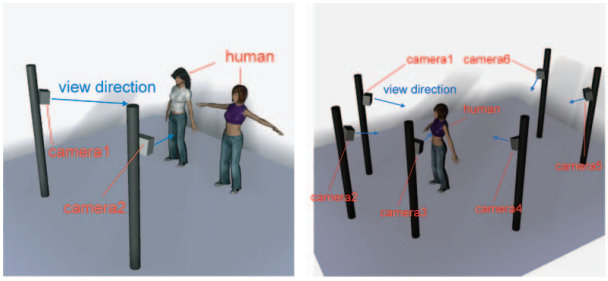


Fig. 5. Motion tracking using the distributed network camera system

This system consists of N network cameras connected to N PCs. In each PC, a 3D human body model is stored individually. All PCs are connected to a main PC by the Internet, in which the integrated posture of the 3D model is stored.

The processing flow is as follows. First, each camera executes the estimation procedure of the posture of 3D model asynchronously, and the compensation angles of joints are estimated in each camera coordinate. Here, the initial posture of the 3D model in each PC is set with the posture stored in the main PC. Next, the obtained compensation angles are sent to the main PC asynchronously, and the posture of the 3D model in the main PC is updated. Due to this asynchronous procedure, the processing cost of each PC is approximately constant, even if the number of cameras increases, and this system is suitable for a distributed multiple-camera system.



(a) Tracking using two cameras (b) Tracking using distributed cameras

Fig. 6. Experimental setup

IV. Motion Tracking Experiment

A. Real-time motion tracking using two cameras

First, we conducted an experiment involving posture estimation using only two cameras. The processing time of the construction of the 2D distance field is 8.7 [ms], and the calculation time of the compensation angles is 3.0 [ms] (Dual-Core Xeon 3.2 GHz, 1 GB). Figure 6(a) shows the experimental condition, and the estimated posture for the seating motion is shown in Fig.7.

B. Real-time motion tracking for two individuals

Next, we conducted an experiment involving simultaneous posture estimation for two people. Figure 8 shows the experimental results. As shown in the figure, simultaneous posture estimation for two individuals was successfully carried out. In this experiment, the processing time of the construction of the 2D distance field was 6.5 [ms], and the calculation time of the compensation angles was 4.2 [ms]. One of the characteristics of the proposed technique is that, once the 2D distance field has been constructed, the processing time is not affected by the number of contours. Therefore, the proposed technique is suitable for the simultaneous motion tracking of multiple individuals.

C. Posture estimation using distributed network cameras

Finally, we conducted an experiment using distributed network cameras. We placed eight cameras in a circle and assumed that these cameras were calibrated manually or by observing mobile robots [27] beforehand. Appropriate cameras are automatically selected according to the distance from the individual and the current posture of this person. More

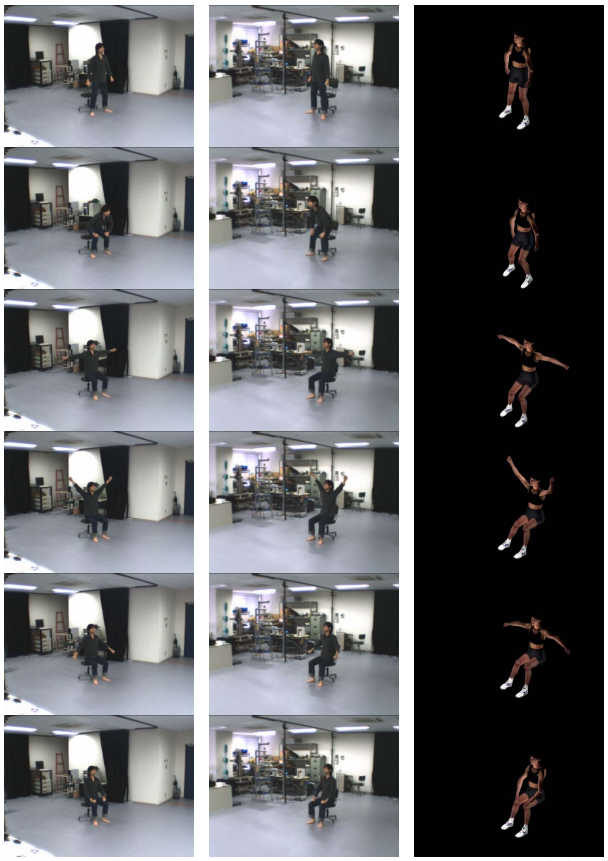


Fig. 7. Tracking results for sitting motion by two cameras

Fig. 8. Tracking results for two people using two cameras

precisely, the posture is estimated by the camera for which the distance to the estimated position of the individual is less than 5 m and the person is facing toward. Figure 6 shows the experimental condition, and Fig.9 shows the experimental results. The processing time for the construction of the 2D distance field and the posture estimation in each PC are 10 [ms] and 7 [ms], respectively, and the processing time for the main PC for one update cycle is 125 [ms].

In this experiment, the individual walks halfway around the circle of cameras in the counterclockwise direction, sits on a chair, stands up, walks halfway around in the clockwise direction, and stands on a chair.

As shown in Fig.9, the appropriate cameras which are shown with light brown are selected automatically according to the position and posture of the individual, and the posture estimation is successfully carried out in any positions of the area.

V. Conclusions

In the present paper, we introduced a real-time motion tracking system that uses distributed network cameras. The proposed system uses the 2D distance field, which is constructed quite rapidly using the silhouette contours by the Fast Marching Method. Unlike the conventional ICP based method, this technique does not require the point correspondence to be determined, and thus fast calculation is possible for the posture estimation. Experimental results obtained using distributed network cameras reveal that various human motions, such as walking, sitting down, and standing up, are estimated by selecting appropriate cameras according to the current situation. Future research will include motion estimation experiments in an everyday environment, such as a house, and collaboration with service robots using estimated human motions.

VI. Acknowledgement

This research was supported in part by the Ministry of Education, Culture, Sports, Science and Technology through a Grant-in- Aid for Scientific Research (B) (No.19360119).

References

[1] T. Hasegawa, R. Kurazume, K. Murakami, and Y. Kimuro, "Robot town platform: a robotic structured environment for daily human life," in *IROS 2008 Workshop on Network Robot System:human concepts of space and activity, integration and application*, 2008.

[2] A. Agarwal and B. Triggs, "3d human pose from silhouettes by relevance vector regression," *IEEE Computer Vision and Pattern Recognition*, vol. 2, pp. 882–888, 2004.

[3] G. Hua, M.-H. Yang, and Y. Wu, "Learning to estimate human pose with data driven belief propagation," *Computer Vision and Pattern Recognition*, vol. 2, pp. 747–754, 2005.

[4] A. Thayananthan, R. Navaratnam, B. S. P. H. Torr, B. Stenger, P. H. S. Torr, and R. Cipolla, "Multivariate relevance vector machines for tracking," in *Proc. European Conference on Computer Vision*. Springer-Verlag, 2005, pp. 124–138.

[5] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 44–58, 2006.
[6] M. Yamamoto, A. Sato, S. Kawada, T. Kondo, and Y. Osaki, "Incremental tracking of human actions from multiple views," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, 1998.
[7] Z. Chen and H. J. Lee, "Knowledge-guided visual perception of 3-d human gait from a single image sequence," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 336–342, 1992.
[8] C. Orrite-Urunuela, J. del Rincon, J. Herrero-Jaraba, and G. Rogez, "2d silhouette and 3d skeletal models for human detection and tracking," in *Proc. of the 17th International Conference on Pattern Recognition*, 2004, pp. 244–247.
[9] L. Campbell and A. Bobick, "Recognition of human body motion using phase space constraints," in *Proc. of IEEE International Conference on Computer Vision*, 1995, pp. 624–630.
[10] D. M. Gavrila and L. S. Davis, "Towards 3-d model-based tracking and recognition of human movement: a multi-view approach," in *International Conference on Automatic Face and Gesture Recognition*, 1995, pp. 272–277.
[11] Q. Delamarre and O. Faugeras, "3d articulated models and multi-view tracking with silhouettes," in *Proc. IEEE International Conference on Computer Vision*, 1999, pp. 716–721.
[12] J. Deutscher, A. Blake, and I.Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. of Computer Vision and Pattern Recognition*, 2000, pp. 2126–2133.
[13] H. Sidenbladh, M. J. Black, and D. Fleet, "Stochastic tracking of 3d human figures using 2d image motion," in *Proc.of Europian Conference on Computer Vision*, 2000, pp. 702–718.
[14] C. Taylor, "Reconstruction of articulated objects from point correspondences in a single uncalibrated image," in *Proc. of International Conference on Computer Vision and Pattern Recognition*, 2000.
[15] J. Carranza, C. Theobalt, M. Magnor, and H. Seidel, "Free-viewpoint of human actors," in *Proc. of SIGGRAPH 2003*, 2003, pp. 569–577.
[16] C. Sminchisescu and B. Triggs, "Kinematic jump processes for monocular 3d human tracking," in *Proc. of International Conference on Computer Vision and Pattern Recognition*, 2004.
[17] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1, pp. 421–428, 2004.
[18] A. Gupta, A. Mittal, and L. S. Davis, "Constraint integration for efficient multiview pose estimation with self-occlusions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 493–506, 2007.
[19] R. Kurazume, H. Omasa, S. Uchida, R. Taniguchi, and T. Hasegawa, "Embodied proactive human interface "pico-2"," in *International Conference on Pattern Recognition*, 2006, pp. B04–0206.
[20] J. Sethian, *Level Set Methods and Fast Marching Methods second ed.* UK: Cambridge University Press, 1999.
[21] C. Sminchisescu and A. Telea, "Human pose estimation from silhouettes: A consistent approach using distance level sets," in *WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, 2002.
[22] Y. Iwashita, R. Kurazume, K. Konishi, M. Nakamoto, M. Hashizume, and T. Hasegawa, "Fast 2d-3d registration for navigation system of surgical robot," in *Proc. IEEE International Conference on on Robotics and Automation*, 2005, pp. 909–915.
[23] R. Kurazume, K. Nakamura, T. Okada, Y. Sato, N. Sugano, T. Koyama, Y. Iwashita, and T. Hasegawa, "3d reconstruction of a femoral shape using a parametric model and two 2d fluoroscopic images," in *Proc. IEEE International Conference on Robotics and Automation*, 2007, pp. 3002–3008.
[24] ——, "3d reconstruction of a femoral shape using a parametric model and two 2d fluoroscopic images," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 202–211, 2009.
[25] Y. Iwashita, R. Kurazume, T. Tsuji, K. Hara, and T. Hasegawa, "Fast implementation of level set method and its realtime applications," in *Proc. IEEE International Conference on Systems, Man and Cybernetics 2004*, 2004, pp. 6302–6307.
[26] R. Smith, "Open dynamics engine," *http://www.ode.org/*.
[27] T. Yokoya, T. Hasegawa, R. Kurazume, and K. Murakami, "Calibration of distributed vision network in unified coordinate system by mobile robots," in *Proc. IEEE International Conference on Robotics and Automation 2008*, 2008, pp. 1412–1417.

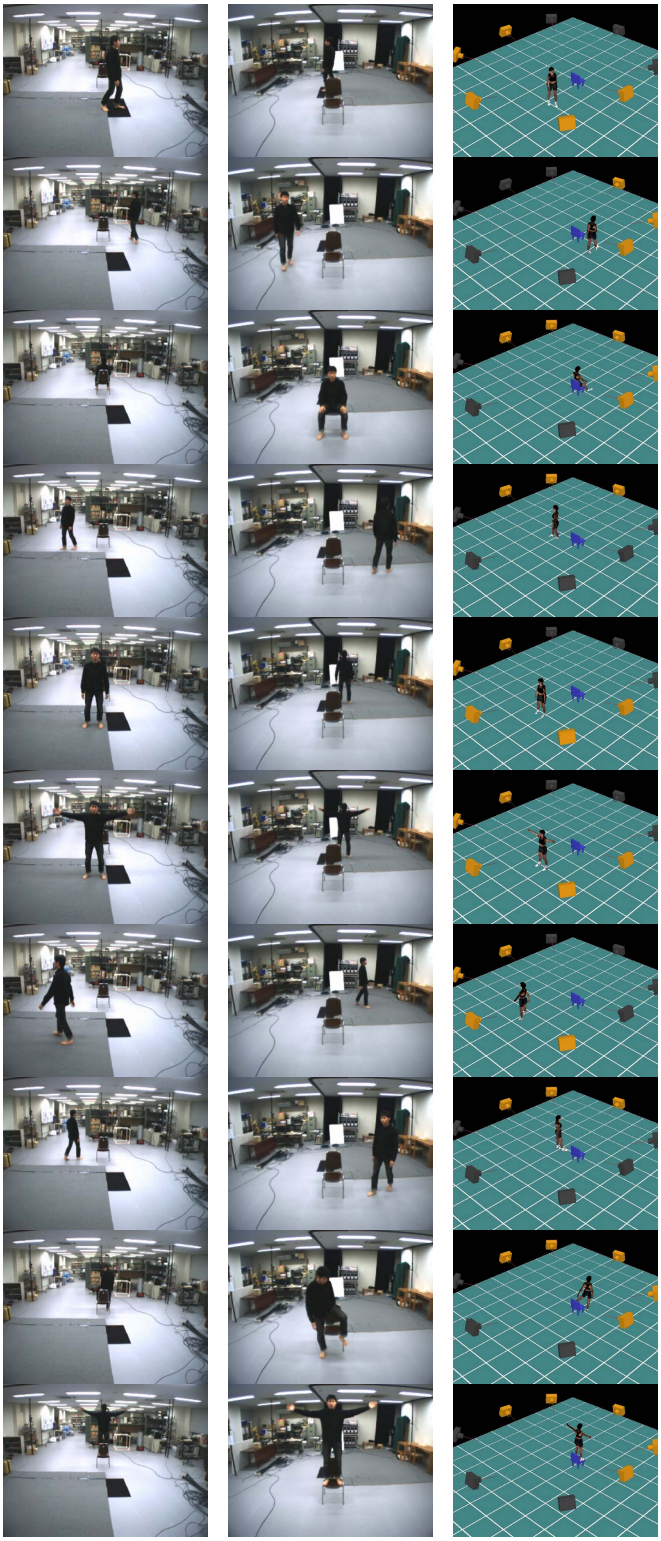


Fig. 9. Tracking results using the distributed camera system