A Hand-Pose Estimation for Vision-Based Human Interfaces

Etsuko Ueda, *Student Member, IEEE*, Yoshio Matsumoto, Masakazu Imai, and Tsukasa Ogasawara, *Member, IEEE*

***Abstract*—This paper proposes a novel method for a hand-pose estimation that can be used for vision-based human interfaces. The aim of this method is to estimate all joint angles. In this method, the hand regions are extracted from multiple images obtained by a multiviewpoint camera system. By integrating these multiviewpoint silhouette images, a hand pose is reconstructed as a "voxel model." Then, all joint angles are estimated using a three-dimensional model fitting between the hand model and the voxel model. The following two experiments were performed: 1) an estimation of joint angles by the silhouette images from the hand-pose simulator and 2) hand-pose estimation using real hand images. The experimental results indicate the feasibility of the proposed algorithm for vision-based interfaces, although the algorithm requires faster implementation for real-time processing.**

***Index Terms*—Hand-pose estimation, model fitting, silhouette image, vision-based human interface, voxel model.**

I. Introduction

HUMAN HAND movements have manipulative functions (e.g., object grasping, object designing) and communicative functions (e.g., sign language, pointing). Most of the "human skills" belong to manipulative function. Since it is difficult to describe manipulative hand movements, teaching human skills to a robot is a hard task. Therefore, we develop a method for measuring quantitative hand poses in real time to enable us to teach human skills to a robot easily and directly.

There are two approaches to measuring a hand pose: contact and noncontact. The former uses contact sensors such as data gloves. The latter uses noncontact sensors such as charge-coupled device (CCD) cameras. In order to build a natural interface for a skill-teaching system, the noncontact approach offers an advantage since it does not restrain the operator's hand movement. In computer vision research, many systems have been developed to recognize predetermined hand poses in real time. Although these system can be used to give commands to a robot, they are not sufficient for teaching skills. Measuring the three-dimensional (3-D) position, the orientation, and all joint angles in real time is necessary. However, since a hand has many degrees of freedom, which causes considerable self-occlusion, no vision-based estimation method for arbitrary hand poses in real time has yet been established.

In this paper, we propose a novel vision-based estimation method for arbitrary hand poses using multiviewpoint silhouette images. The remainder of this paper is organized as follows. Section II outlines previous proposed methods. Section III describes the hand representation used in our system. Section IV presents the details of the hand-pose estimation algorithm. Section V describes the results of experiments using the hand-pose simulator. Section VI describes the results of experiments using a real camera system. Finally, in Section VII, we discuss our results and describe future work.

Manuscript received December 25, 2001; revised August 17, 2002. Abstract published on the Internet May 26, 2003. This work was supported in part by the Ministry of Education, Science, Sports and Culture under Grant-in-Aid for Exploratory Research 14658115. This paper was presented at the 2001 IEEE International Workshop on Robot and Human Interactive Communication, Bordeaux and Paris, France, September 18–21.

E. Ueda, Y. Matsumoto, and T. Ogasawara are with the Robotics Laboratory, Nara Institute of Science and Technology, Nara 630-0192, Japan (e-mail: etsuko-u@is.aist-nara.ac.jp; yoshio@is.aist-nara.ac.jp; ogasawar@is.aist-nara.ac.jp).

M. Imai is with the Department of Information Systems, Tottori University of Environmental Studies, Tottori 689-1111, Japan (e-mail: imai@kankyo-u.ac.jp).

Digital Object Identifier 10.1109/TIE.2003.814758

II. Related Research

Previously proposed methods of vision-based hand-pose estimation can be classified into the following two categories:

- estimation of communicative hand poses [1]–[3]
- estimation of manipulative hand poses [4]–[7]

The former includes hand-pose recognition systems for sign language and hand-shape recognition systems for virtual reality (VR) interfaces. Utsumi *et al.* used multiviewpoint images to manipulate objects in the virtual world [2]. Eight kinds of commands were recognized based on the shape and movement of the hands. In the Gesture Computer developed by Maggioni, the shape of a hand was recognized based on computing the moment of a hand silhouette image and detecting the fingertips [3]. In this research, hand-shape recognition in real-time is possible. However, only predetermined hand shapes can be recognized and used as commands.

In contrast, estimation of manipulative hand poses targets arbitrary hand poses. Shimada *et al.* performed hand-pose estimation from a monocular image sequence based on loose constraints [4]. Kameda *et al.* performed hand-pose estimation from a monocular silhouette image using a two-dimensional model matching of the image and an articulated object model [5]. However, since depth information cannot be obtained from a monocular image, it is difficult to estimate accurate hand poses using a single camera. Delamarre *et al.* proposed a hand-pose estimation method using a stereo image pair. The virtual forces generated between a hand model and reconstructed surface data are used for 3-D model fitting [6]. In the DigitEyes system by Rehg [7], the tracking of a 27 degrees-of-freedom (DOFs) hand model can be performed in about 10 Hz using two cameras. In a stereo camera system, depth information can be obtained by stereo matching. However, inevitable mismatching results in a

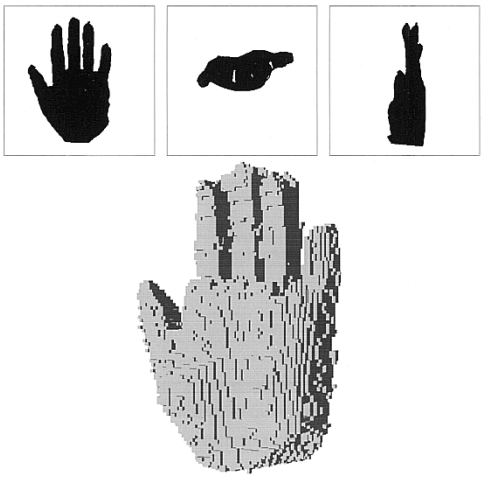


Fig. 1. Reconstruction of a voxel model (upper: silhouette images; lower: voxel model).

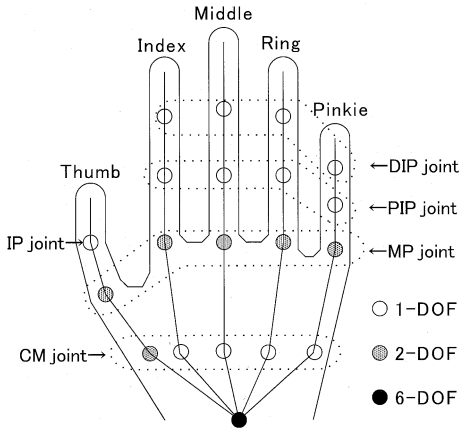


Fig. 2. Skeletal hand model.

deterioration in accuracy. Neither the single-camera system nor the stereo-camera system has the necessary occlusion tolerance because the 3-D information is insufficient.

However, two advantages emerge in the multiviewpoint camera system, when compared with the monocular and the stereo camera systems. These advantages include the following.

- The influence of self-occlusion is smaller than that in the monocular and the stereo camera systems.
- The 3-D shape can be reconstructed from the multiviewpoint images.

Using these advantages, complicated 3-D shape reconstruction from multiviewpoint images has recently become an intensively researched area. Szeliski [8] proposed an efficient algorithm for volume reconstruction via a visual cone intersection. Saito [9] presented a multicamera approach for the volume reconstruction in project grid space. However, Szeliski and Saito were not concerned with performing their methods in real time. Davis [10] and Tokai [11] proposed methods to perform Szeliski's algorithm in real time using PC clusters. Dyer [12] proposed a voxel-coloring technique which can reconstruct the photo-realistic 3-D volume data by coloring the voxel using color images. However, the goal of these works is simply 3-D shape reconstruction of objects.

Our system uses the multiviewpoint camera system for the observation of hand poses. The obtained silhouette images are converted to 3-D volumetric data by Szeliski's method. Next, the hand pose is estimated by a 3-D model fitting using reconstructed 3-D volumetric data.

III. Representation of the Hand

A. Voxel Model

The 3-D shape reconstruction method in this research is equivalent to a "shape from silhouette." The reconstructed 3-D shape using "octree representation" is utilized as the observational data of a hand. This observational data is termed a "voxel model." The accuracy of the voxel model can be changed with the number of hierarchies of the tree. The method of constructing a voxel model is identical to the method described in [8]. In the reconstruction process, we regard on octant, which lies on the boundary of the hand as an octant inside of the hand. The voxel model is created to circumscribe the actual fingers. Consequently, our voxel model is slightly larger than the actual hand. Fig. 1 shows a voxel model that was reconstructed from three silhouette images.

B. Hand Model

In this research, the 3-D hand model consists of: 1) a skeletal hand model and 2) a surface hand model. This is basically the same model as the one proposed by Yasumuro *et al.* [13].

1) Skeletal Hand Model: A hand is modeled as a set of five manipulators that have a common base point at the wrist. Each finger is represented as a set of links and joints as shown in Fig. 2. The hand posture can, thus, be represented using a kinematic model of the manipulators. This model of the hand is called the "skeletal hand model." It has 31 DOFs in total, including the translation and the rotation of the wrist. The first joints of the index, the middle, the ring, and the pinkie are called the DIP joint. The second joints of the index, the middle, the ring, and the pinkie are called the PIP joint. The third joints of the index, the middle, the ring, the pinkie, and the second joint of the thumb are called the MP joint.

2) Surface Hand Model: When all of joint angles are determined, the posture of the hand can be determined. In order to render the image of a hand, the surface data of the hand's skin are needed. The shape of the hand surface must deform according to the skeletal posture. For that purpose, the shape of the hand surface is represented by triangular patches, and each vertex of the triangular patch has an attribute that indicates the corresponding skeletal link.

IV. Hand-Pose Estimation

A. Outline of the Proposed Method

Each joint angle is estimated by fitting the surface hand model to the voxel model. Two approaches exist for model fitting using two-dimensional (2-D) observed data and a 3-D model.

1) The 3-D model is projected onto a 2-D plane, and the model fitting is done in the 2-D plane (conventional approach).

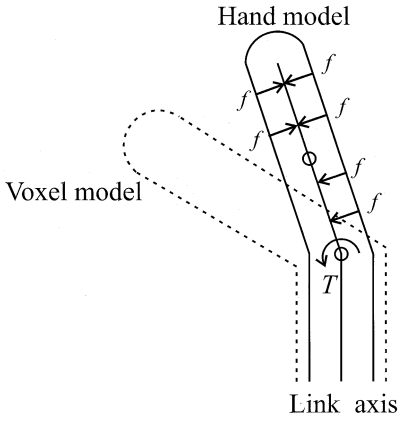


Fig. 3. Scheme of model fitting.

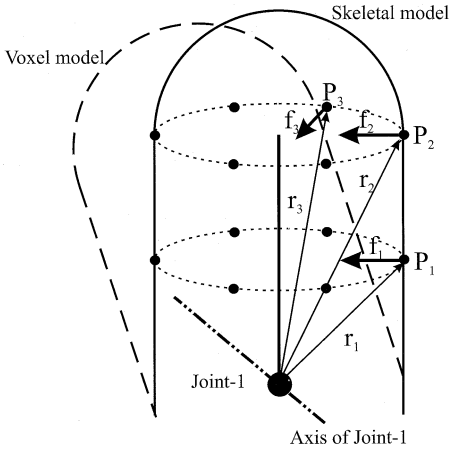


Fig. 4. Generation of torque.

2) The 3-D shape is reconstructed by combining the 2-D observed data, and the model fitting is done in 3-D space (our approach).

Our method belongs to the latter approach 2), which directly handles the 3-D deformation of the shape of the model. In approach 1), the process that determines which finger corresponds to which part of the conventional data is required. However, in our approach, only the geometric information of the hand model and the voxel model are used for model fitting without any heuristic or a priori information. From this point of view, our approach is simpler than that of conventional methods. In addition, the simple algorithm described in this paper is suitable for parallel processing.

The voxel model represents the area that a hand occupies in the voxel space. The surface hand model also represents the position where the hand exists in terms of the vertex coordinates of the triangular patches. When the surface hand model is completely included in the voxel model, the skeletal hand model fits the observational data.

When the angle of a joint is represented as $\mathbf{a}_i = \{a_i(k)|0 \leq k < 3\}$ ($a_i(k)$ is the joint angle of the ith joint in axis k), the hand posture can be represented as $P = \{\mathbf{a}_i|0 < i < r\}$ (r is the number of all joints). In this posture, the coordinates of the vertices that constitute the surface hand model are defined as $L = \{\mathbf{p}(m)|0 \leq m < q\}$ ($\mathbf{p}(m)$ is the vertex coordinate, q is the number of vertices). Each $\mathbf{p}(m)$ is determined by P. V is then defined as the occupied area of a voxel model. The hand-pose estimation is now a process to find a P that satisfies $L \subset V$. P that realizes $Out = 0$ is determined under the evaluation function $Out = \{\mathbf{p}(m) \not\subset V|0 \leq m < q\}$. For this purpose, a force vector that makes the skeletal hand model approach the voxel model is generated for each point included in Out. This process is iteratively performed while changing the joint angles gradually by the generated force vector.

B. Details of the Estimation Algorithm

The details of the estimation are described as follows.

Step 1) Silhouette images are created using images captured by a multiviewpoint camera system.

Step 2) The voxel model is created from the set of these silhouette images.

Step 3) The skeletal hand model representing the hand pose is compared with the voxel model representing the observed hand shape. The vertices of the triangular patches located outside the voxel model are marked on.

Step 4) Force f which includes a direction to a joint axis, is generated in the focused vertex as shown in Fig. 3. The generated force f is then converted to a torque around the joint, which is termed t. t is summed up, and the total torque T is determined.

Step 5) The joint angle is changed by $\Delta\alpha$ according to the direction of the torque.

Step 6) The joint positions are recalculated from the new joint angle and the coordinates of the vertices in the surface hand model are updated.

Step 7) The evaluation function is calculated. If the result of the evaluation is below a threshold, the estimation finishes. Otherwise, the process goes back to Step 3).

The determination of the rotative direction of each joint in Step 4) is described in Fig. 4. Vertices P_1, P_2, and P_3 are located outside the voxel model. It is given beforehand that these vertices are related to the rotation of Joint-1. Next, perpendicular forces to a joint axis are generated to each vertex described as $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$. The vectors from the position of a related joint (Joint-1) to these vertices are defined as $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$. Then, torque $\mathbf{t}_i$ in these vertices is calculated by the distance vector $\mathbf{r}_i$ and the force vector $\mathbf{f}_i$. The torque of each of these vertices is totaled up to produce $\mathbf{T}$, which is the rotation torque of Joint-1. The rotative direction in the axis of Joint-1 is determined using the direction of torque $\mathbf{T}$. In the case of Fig. 4, Joint-1 is rotated $+\Delta\alpha^\circ$ degrees. In Step 7), the calculation of the convergence ratio is performed for each joint. The convergence ratio is defined as follows:

$$rate = \frac{in\_vertex}{all\_vertex} \times 100(\%)$$

where $rate$ is the convergence ratio, in_vertex is the number of V's that are located inside the voxel model, and all_vertex is the number of V's which are related to the rotation of the focused joint. When all of the vertices that are related to the rotation of the focused joint are located inside the voxel model, the angle estimation of the focused joint is completed. However,

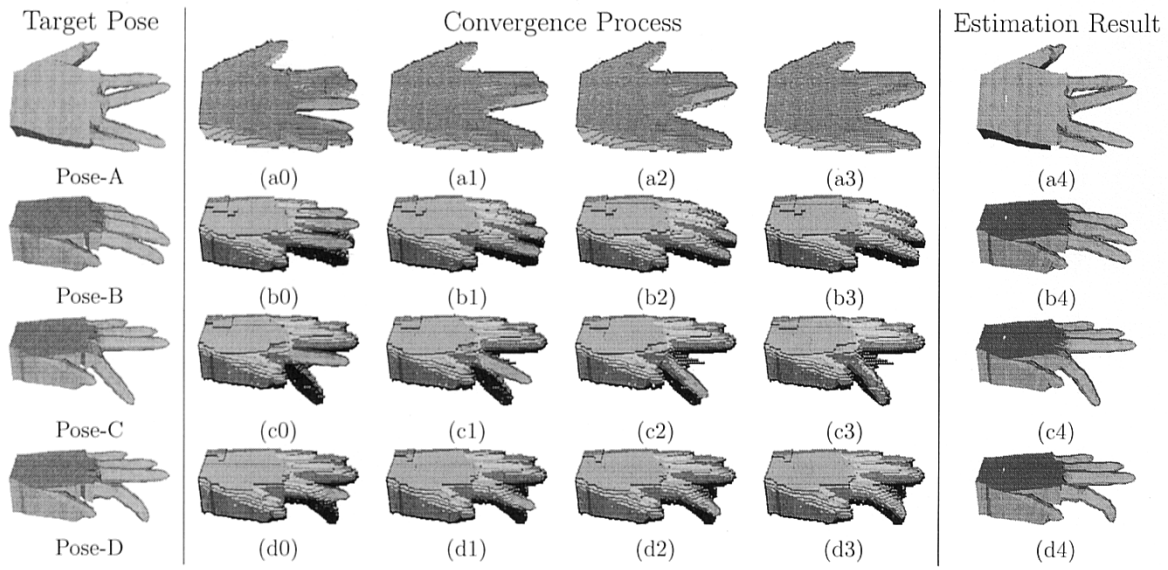


Fig. 5. Convergence process.

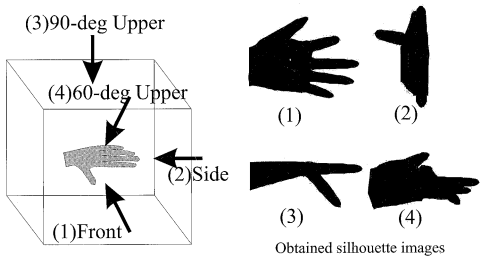


Fig. 6. Viewpoints and obtained silhouette images.

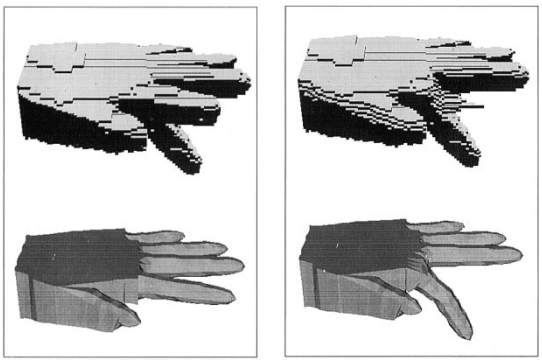


Fig. 7. Effect of camera position and number of cameras (upper: voxel model (level = 8); lower: estimated pose). (a) Three cameras. Viewpoints 1, 2, and 3. (b) Four cameras. Viewpoints 1, 2, 3, and 4.

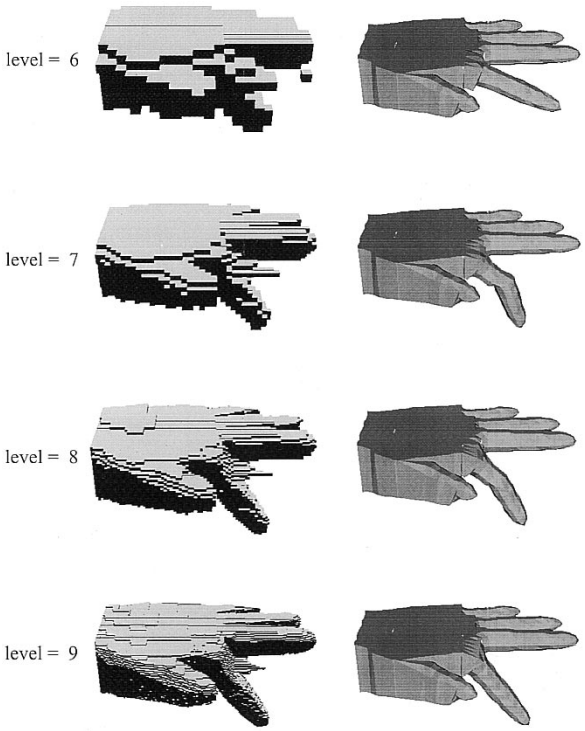


Fig. 8. Effect of the octree level (left: voxel model; right: estimated pose).

when an estimated angle exceeds the movable range of the joint, or the direction of torque begins to vibrate, the angle estimation of the finger is terminated.

V. Hand-Pose Estimation Using a Simulator

A. Estimation Results

We generated and estimated various kinds of hand poses using a hand-pose simulator. The simulation system has a dual-Pentium III 1-GHz system, and the memory size is 1024 MB. In Fig. 5, Pose-A–Pose-D show four examples of the generated hand poses. The hand poses are as follows.

- Pose-A: The MP joint of the index finger is adducted by 8°. The MP joint of the middle finger is abducted by 15°. The MP joint of the ring finger is adducted by 10°. The MP joint of the pinkie is abducted by 5°.
- Pose-B: The MP joint, the PIP joint, and the DIP joint of the index, the middle, the ring, and the pinkie are respectively flexed by 10°.
- Pose-C: The MP joint of the index is flexed by 50°.

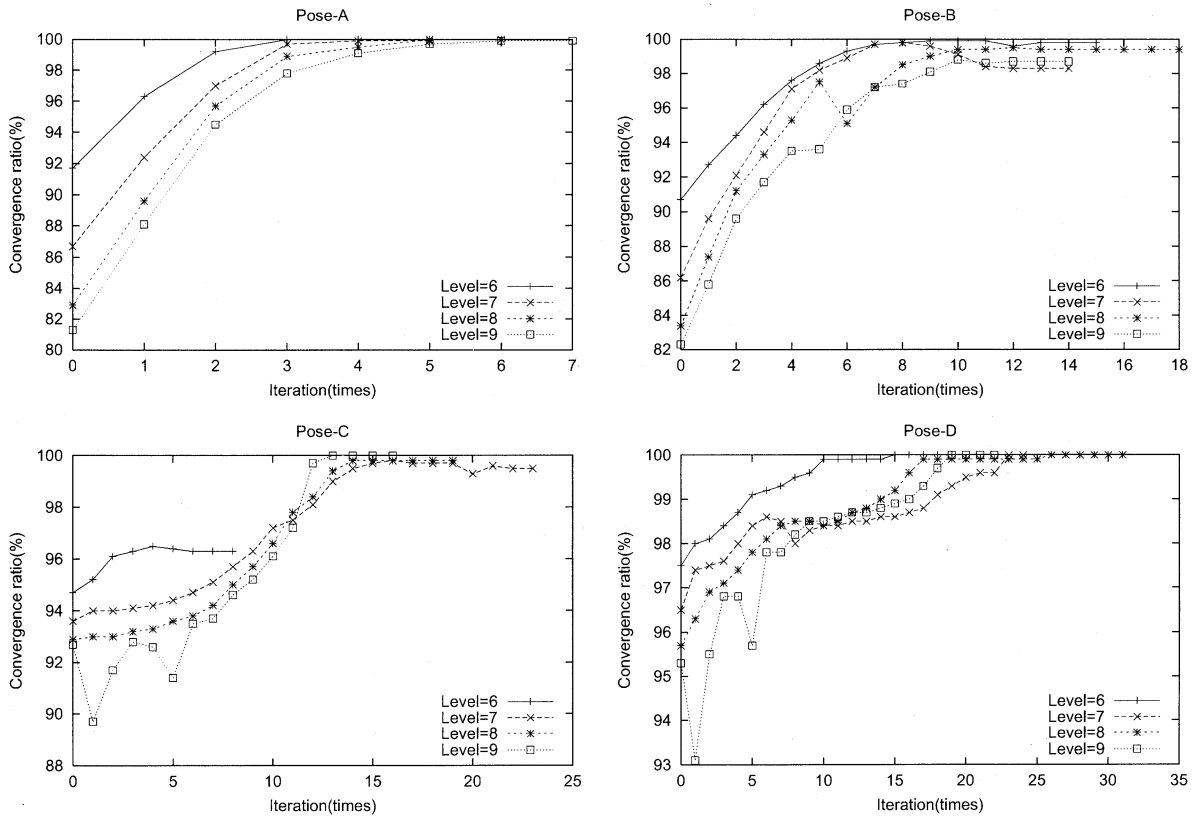


Fig. 9. Convergence ratio.

Pose-D: The MP joint of the index is abducted by 10°. The PIP joint and the DIP joint of the index are respectively flexed by 30°.

The purpose of our method is continuous and arbitrary hand-pose estimation in real time. Therefore, we assume that the shape change between consecutive frames is small. Concretely, we assume that the change of each joint is at most 15° per frame. From this point of view, Pose-A and Pose-B are examples of the pose change assumed in this paper. Although Pose-C and Pose-D have larger changes, we also performed the simulation using these poses in order to confirm the robustness of our method.

The estimated results of these hand poses are shown in Fig. 5. In this figure, (a0), (b0), (c0), and (d0) show the initial hand model superimposed on the voxel model, and (a1), (b1), (c1), and (d1)–(a3), (b3), (c3), and (d3) show the convergence processes of the surface hand model to the voxel model. (a4), (b4), (c4), and (d4) show the final estimated poses of the hand model, which are completely included in the voxel model. By comparing (a4), (b4), (c4), and (d4) with the generated poses (Pose-A–Pose-D), it is clear that the estimations were performed correctly for all poses.

B. Effect of Camera Position and the Number of Cameras

The camera position and the number of cameras greatly affect the accuracy of the voxel model. An experiment was conducted to confirm the effect of the camera position. Fig. 6 shows the camera positions and the obtained silhouette images, when observing the hand pose of Pose-C.

The result of the voxel generation using three cameras (numbers 1, 2, and 3) is shown in Fig. 7(a). Since an incorrect index finger is generated due to occlusion, the estimated pose of the index finger is wrong. After adding camera number 4, the incorrect index finger in the reconstructed voxel model almost disappeared and the estimated pose of the index finger become correct as shown in Fig. 7(b). This experiment indicates that the number and configuration of the cameras are important in generating a correct voxel model of a hand.

C. Effect of the Octree Level

The accuracy of the estimation depends on the number of hierarchies for the octree. We use the term “octree level” as the number of hierarchies in this paper. Fig. 8 (left) shows the experimental results of voxel generation at octree level 6, 7, 8, and 9. The corresponding minimum octant of the voxel model is 8-, 4-, 2-, and 1-mm cube, respectively. As mentioned in Section III-A, the voxel model is created to circumscribe the actual fingers.

A quantitative experiment was conducted to confirm the relationship between the octree level and the accuracy of the estimation for four poses (Pose-A–Pose-D). Fig. 8 (right) shows the experimental results of the estimation for Pose-C with octree levels 6, 7, 8, and 9.

Fig. 9 shows the profiles of the convergence ratio during the estimation process. With a minor change of hand pose (Pose-A and Pose-B), the convergence is quick when the octree level is low. The reason for this is that the size of the voxel model tends to be larger than the size of an actual hand. For a major change in

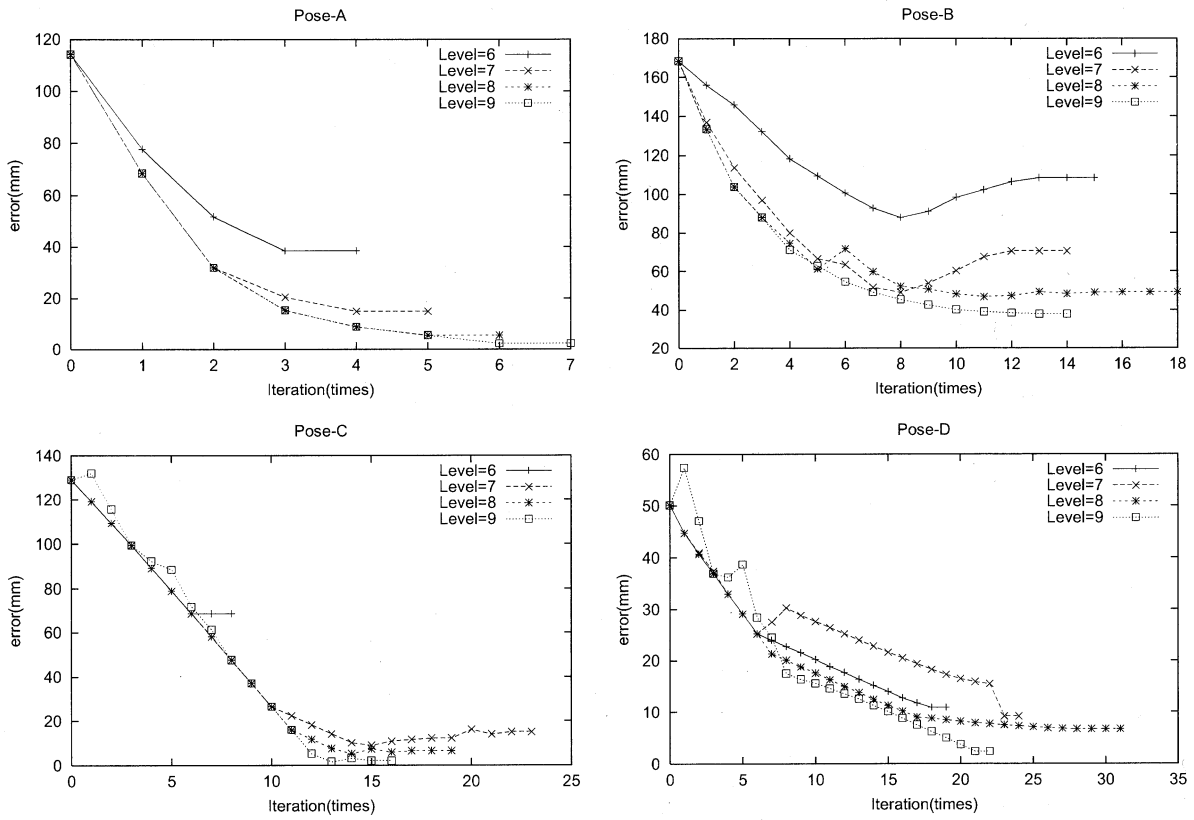


Fig. 10. Estimation error.

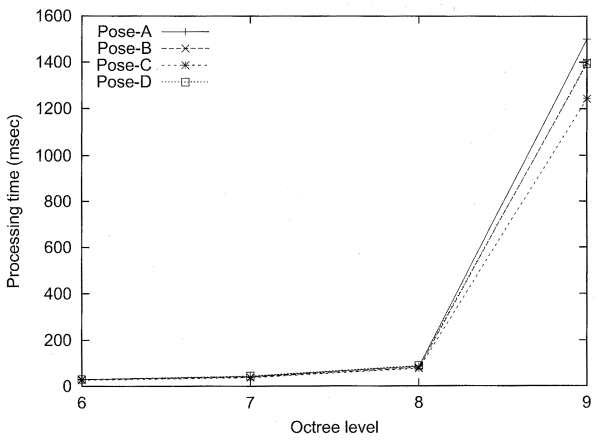


Fig. 11. Processing time of voxel creation.

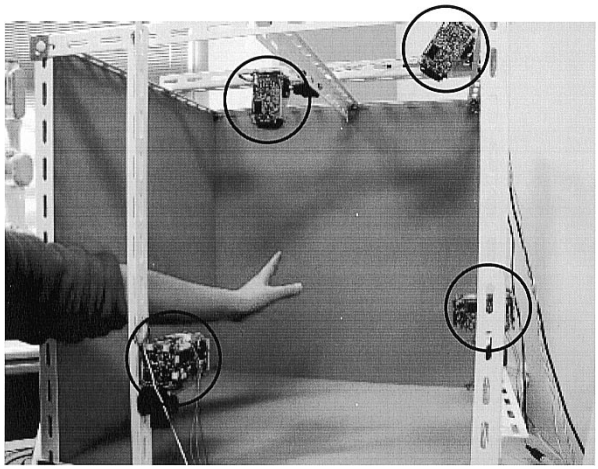


Fig. 12. Experiment environment.

hand pose (Pose-C and Pose-D), the low octree level sometimes leads to an incorrect estimation (Pose-C, Level 6).

The convergence ratio only provides an indication of the termination of the convergence process, and does not necessarily reflect the accuracy of the estimation. The reason for this is because the girth of the fingers in the voxel model differs depending on the octree level as shown in Fig. 8 (left). When the octree level is low, the voxel model tends to be large and there may be large estimation errors left even with the convergence ratio of 100%. On the other hand, when the octree level is high, the voxel model becomes precise, and the convergence ratio rarely reaches 100%, even when the estimation is correct.

In order to evaluate the accuracy of estimation, the estimation error is defined as follows:

$$error = \sum_i dist(epos_i, tpos_i)$$

where $epos_i$ is the estimated position of the ith joint, $tpos_i$ is the target position of ith joint, and $dist(epos_i, tpos_i)$ is the Euclidean distance between $epos_i$ and $tpos_i$. Fig. 10 shows the profiles of the errors with octree levels 6–9. It can be seen that the accuracy in the estimation improves as the octree level becomes higher for all poses.

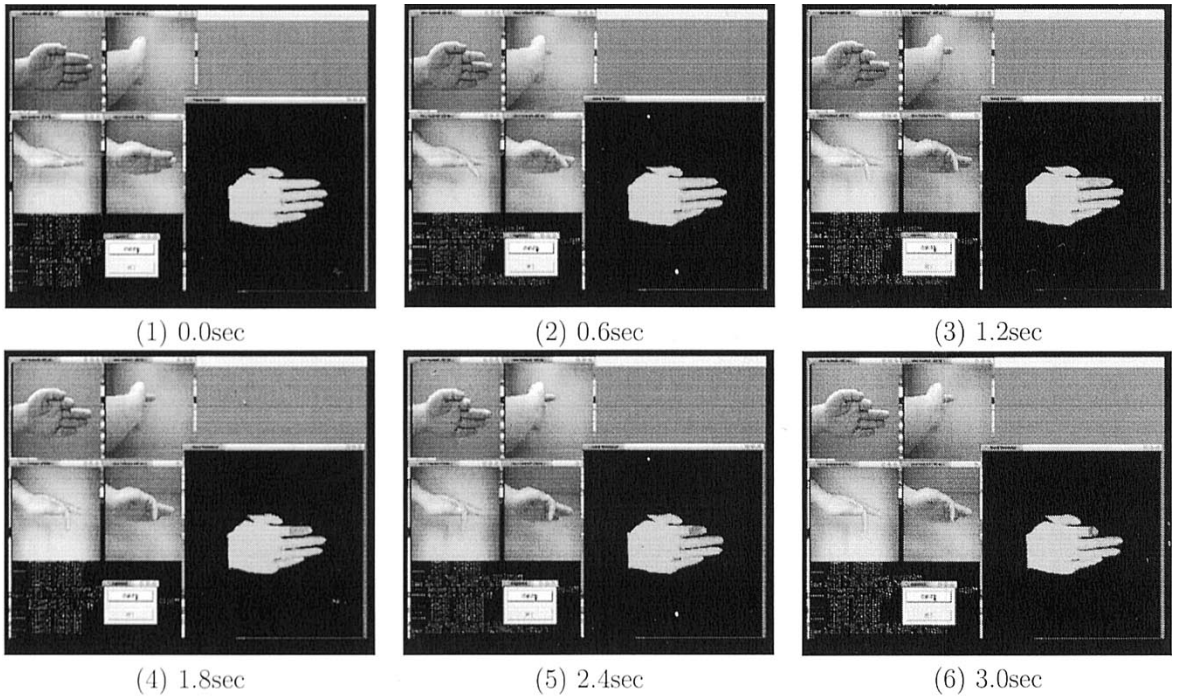


Fig. 13. Hand-pose estimation using real images (1).

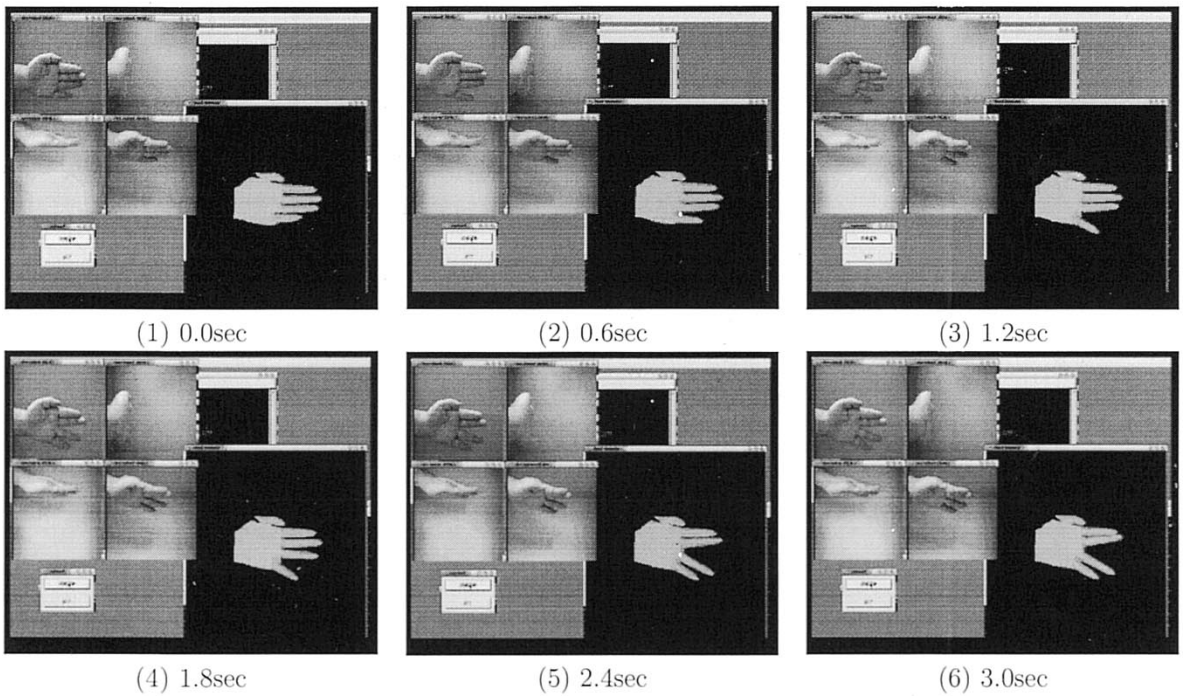


Fig. 14. Hand-pose estimation using real images (2).

Fig. 11 shows the processing time of the voxel model creation for every level. From levels 6 to 9, the processing time increases as the octree level becomes higher. When the octree level is 9, the processing time drastically increases. The reason is as follows. We precompute a lookup table to project each octant to the image planes. However, when the octree level is 9, the whole projection table cannot be created due to the limitation of memory.

These figures (Fig. 9–11) indicate that the octree level influences both the accuracy of the estimation and the convergence speed. Therefore, the octree level should be determined according to the desired accuracy and processing speed.

TABLE I
PROCESSING TIME

Pre-processing	130 msec
Construction of Voxel Model	50 msec
Pose Estimation	160 msec
Total	340 msec

VI. HAND-POSE ESTIMATION USING REAL IMAGES

A. Experimental System

In order to estimate hand poses using real hand images, an actual camera system was built. We constructed an estimation space with a 60-cm cube. Four CCD cameras were mounted on the frame to capture hand images. The cameras were positioned in the front, on the side, in the upper part, and 45° above the center of the work space, respectively. However, the number of cameras can be changed according to the required processing speed and accuracy. Fig. 12 shows the overview of the experimental environment.

At present, blue boards are installed as the background in order to facilitate the generation of the silhouette images. By incorporating the techniques described in [14], we will be able to perform the silhouette extraction of a hand without this artificial background.

B. Result of Experiment

Figs. 13 and 14 show the results of the online estimation using real images. Fig. 13 shows the pose of only the index finger bending forward. Fig. 14 shows the pose with the abducting fingers. These experimental results indicate that the proposed method works successfully with real images, although the processing speed is still slow.

The processing time for our current system is shown in Table I. The preprocessing includes image binarization, noise reduction, and making a half-distance map [8]. The number of cameras is four, and the maximum level of the octree is 7. The system has a dual-Pentium III 1-GHz system which is the same as in the simulation system.

VII. CONCLUSION

In this paper, we proposed a novel hand-pose estimation method, which can be used for vision-based interfaces. In the method, a hand is represented by a hand model consisting of link data and surface data. A voxel model is reconstructed from silhouette images of the hand obtained from a multiviewpoint camera system. The joint angles of the skeletal hand model are estimated by fitting the surface hand model to the obtained voxel model. In order to confirm the feasibility of the proposed method, we generated various kinds of hand poses using a hand-pose simulator, and estimated the hand poses. Finally, experiments using real images were conducted. The proposed method can be applied not only to skill teaching systems but also to interfaces for virtual reality and 3-D design systems.

At present, there are two major problems in our system. One is an accuracy problem, which is caused by modeling errors in the hand model. A technique to model hands more accurately is required. The other problem is processing speed. To build natural and practical human interfaces, the system should run at least in 10 Hz, which is three times faster than our current system.

As future research, we are planning to overcome these weak points, and to build a natural interface for a skill teaching system using our method.

REFERENCES

[1] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: a review," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 677–695, July 1997.

[2] A. Utsumi, J. Ohya, and R. Nakatsu, "Multiple-hand-gesture tracking using multiple cameras," in *Proc. IEEE Computer Society Conf. Computer Vison and Pattern Recognition*, 1999, pp. 473–478.

[3] C. Maggioni and B. Kämmerer, "Gesture computer—history, design and applications," in *Computer Vision for Human-Machine Interaction*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[4] N. Shimada, Y. Shirai, and Y. Kuno, "3-D pose estimation and model refinement of an articulated object from a monocular image sequence," in *Proc. 3rd Conf. Face and Gesture Recognition*, 1998, pp. 268–273.

[5] Y. Kameda, M. Minoh, and K. Ikeda, "Three dimensional pose estimation of an articulated object from its silhouette image," in *Proc. Asian Conf. Computer Vision*, 1993, pp. 612–615.

[6] Q. Delamarre and O. Faugeras, "Finding pose of hand in video images: a stereo-based approach," in *Proc. 3rd Conf. Face and Gesture Recognition*, 1998, pp. 585–590.

[7] J. M. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: an application to human hand tracking," in *Proc. European Conf. Computer Vision*, vol. 2, 1994, pp. 35–46.

[8] R. Szeliski, "Rapid octree construction from image sequences," *CVGIP, Image Understanding*, vol. 58, no. 1, pp. 23–32, July 1993.

[9] H. Saito and T. Kanade, "Shape reconstruction in projective grid space from large of images," in *Proc. IEEE Computer Society Conf. Computer Vison and Pattern Recognition*, 1999, pp. 49–54.

[10] L. Davis, E. Borovikov, R. Culter, D. Harwood, and T. Horprasert, "Multi-perspective analysis of human action," in *Proc. Third Int. Workshop Cooperative Distributed Vision*, 2000, pp. 189–223.

[11] S. Tokai, T. Wada, and T. Matsuyama, "Real time 3D shape reconstruction using PC cluster system," in *Proc. Third Int. Workshop Cooperative Distributed Vision*, 2000, pp. 171–187.

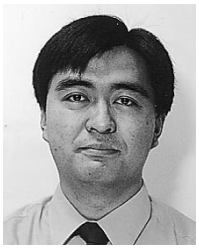
[12] C. R. Dyer, "Volumetric scene reconstruction from multiplue views," in *Foundation of Image Analysis*. Boston, MA: Kluwer, 2001.

[13] Y. Yasumuro, Q. Chen, and K. Chihara, "Three-dimensional modeling of the human hand with motion constraints," *Image Vis. Comput.*, vol. 17, no. 2, pp. 149–156, 1999.

[14] D. Snow, P. Viola, and R. Zabih, "Exact voxel occupancy with graph cuts," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2000, pp. 345–352.

Etsuko Ueda (S'00) received the B.E. degree in 1999 from the National Institution for Academic Degrees (NIAD), Tokyo, Japan, and the M.E. degree in engineering in 2001 from Nara Institute of Science and Technology, Nara, Japan, where she is currently working toward the Ph.D. degree.

Her research interests include vision-based human interfaces.



Yoshio Matsumoto received the B.E., M.E., and Dr. of Engineering degrees from The University of Tokyo, Tokyo, Japan, in 1993, 1995, 1998 respectively.

In 1998, he joined the Australian National University as a Research Fellow, where he developed a real-time vision system for gaze measurement. In 1999, he moved to the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan, as a Research Associate. In 2002, he became an Associate Professor. His research interests include real-time vision processing technology which can be applied to mobile robot navigation and human interfaces.

Masakazu Imai received the B.S., M.S., and Ph.D. degrees from Osaka University, Osaka, Japan, in 1982, 1984, and 1987, respectively.

He is a Professor in the Department of Information Systems, Tottori University of Environmental Studies, Tottori, Japan. Prior to joining Tottori University of Environmental Studies in 2001, he was an Associate Professor in the Graduate School of Information, Nara Institute of Science and Technology, Nara, Japan. His research interests are in the area of media processing, including computer vision and pattern recognition. He is also interested in the research area of digital libraries.

Tsukasa Ogasawara (M'88) was born in Ehime, Japan, in 1955. He received the B.E. degree in measurements and mathematical engineering and the M.E. and Ph.D. degrees in information engineering from The University of Tokyo, Tokyo, Japan, in 1978, 1980, and 1983, respectively.

From 1983 to 1998, he was with the Electrotechnical Laboratory, Ministry of International Trade and Industry, Japan. From 1993 to 1994, he was with the Institute for Real-Time Computer Systems and Robotics, University of Karlsruhe, Germany, as a Humboldt Research Fellow. He joined Nara Institute of Science and Technology, Nara, Japan, in 1998, and is currently a Professor in the Graduate School of Information Science. His research interests include programming environments, man–machine interfaces, and computer architectures for intelligent robots.