Hand Pose Estimation for Vision-based Human Interface

Etsuko Ueda[†] Yoshio Matsumoto[†‡] Masakazu Imai[†] Tsukasa Ogasawara[†]
[†]Nara Institute of Science and Technology.
8916-5, Takayama-cho, Ikoma, Nara, Japan.
[‡]CREST, JST(Japan Science and Technology Corporation)
E-mail {etsuko-u,yoshio,imai,ogasawar}@is.aist-nara.ac.jp

Abstract

This paper proposes a novel method for hand pose estimation that can be used for vision-based human interfaces. The aim of the method is to estimate all joint angles to manipulate an object in the virtual space. In this method, the hand regions are extracted from multiple images obtained by the multi-viewpoint camera system. By integrating these multi-viewpoint silhouette images, a hand pose is reconstructed as a "voxel model". Then all joint angles are estimated using three dimensional model fitting between hand model and voxel model. An experiment was performed in which the joint angles were estimated from the silhouette images by the hand-pose simulator. The experimental result indicates the feasibility of the proposed algorithm for vision-based interfaces.

1 Introduction

Besides its normal use for handling objects and manipulating tools, the hand of human being can be used as the mean of communication. For example, people can express their feelings by gestures, and even "talk" by the sign language. It is an effective approach to make use of the hand movement that has abundant powers of expression as a natural human-robot or human-computer interaction. For that purpose, it is necessary to recognize hand movements of a user in real-time.

There are two kind of methods for hand pose estimation : contact and non-contact ones. The former one uses contact sensors such as data gloves. The latter uses non-contact sensors such as CCD cameras. Until now, many researches have been done on vision-based hand pose estimation. However, the estimation method for arbitrary hand poses in real-time is not yet established.

This paper proposes a novel method that can be used as a vision-based human interface for arbitrary hand pose estimation. The remainder of this paper is organized as follows. Section 2 mentions about the previous proposed method. Section 3 describes the representation of a hand that is used at our system. Section 4 describes the detail of hand pose estimation algorithm. Section 5 describes the result of experiment using the hand-pose simulator. Finally, we discuss the results and describe the future work.

2 Related Research

Previously proposed methods of vision-based hand pose estimation are classified into two categories.

- Estimation of communicative hand poses[1, 2, 3]
- Estimation of manipulative hand poses[4, 5, 6]

The former includes hand pose recognition systems for the sign language and a hand shape recognition system for the VR interfaces. Utsumi et al. used multi-viewpoint images to perform the operation of objects in the virtual world[2]. Eight kinds of commands are recognized based on the shape and movement of the hands. In Gesture Computer developed by Maggioni, the shape of a hand was recognized based on computing the moment of a hand silhouette image and detecting finger tips[3]. In these researches, the hand shape recognition in real-time is possible. However, only predetermined hand shapes can be recognized and used as commands.

In contrast, the latter deals with 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 two-dimensional model matching of the image and articulated object model[5]. However, since the depth information cannot be obtained from a monocular image, it is difficult to estimate the accurate hand pose using a single camera.

Delamarre et al. proposed the hand pose estimation method using a stereo image pair. The virtual forces generated between a hand model and a reconstructed

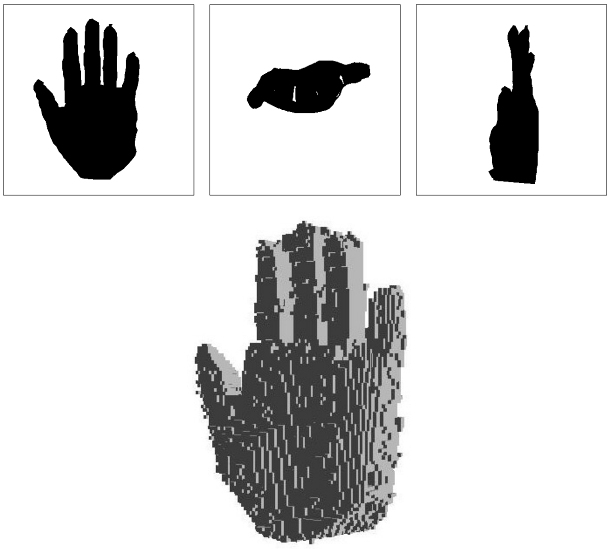


Figure 1: Reconstruction of voxel model
(upper : silhouette images, lower : voxel model)

surface data is used for 3D model fitting[6]. In a stereo camera system, depth information can be obtained by stereo matching. However, inevitable mismatching results in deterioration in accuracy.

In the multi-viewpoint camera system, the influence of self-occlusion can be smaller than the monocular and stereo camera systems. Furthermore, 3D shape can be reconstructed as volumetric data from multi-viewpoint silhouette images[7, 8], which is more stable than the depth information obtained by the stereo matching. In this research, the hand pose is estimated using the reconstructed 3D volumetric data.

3 Representation of Hand

3.1 Voxel Model

The 3D shape reconstruction method in this research is equivalent to the “shape from silhouette”. The reconstructed 3D shape using “octree representation” is dealt as the observational data of a hand. The observational data is termed the “voxel model”. The accuracy of the voxel model can be changed with the number of hierarchies for the tree. The method of constructing a voxel model is identical to the method described in [8]. Figure 1 shows the voxel model that was reconstructed from three silhouette images.

3.2 Hand Model

In this research, the 3D hand model consists of 1) the skeletal hand model and 2) the surface hand model. This is basically the same model as the one proposed by Yasumuro et al.[9].

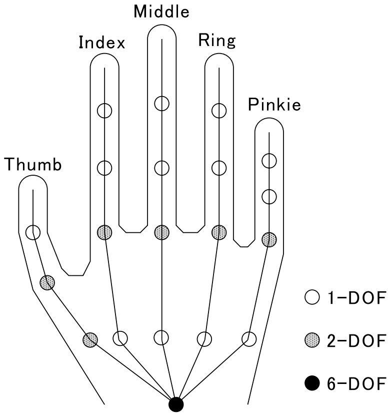


Figure 2: Skeletal hand model

3.2.1 Skeletal Hand Model

A hand is modeled as a set of five manipulators that have a common base point at a wrist. Each finger is represented as a set of links and joints as shown in Figure 2. Thereby, the hand posture can be represented using the kinematic model of the manipulators. This model of the hand is called the “skeletal hand model”. In order to represent various hand pose, the degrees of freedom of each link has been arranged as shown in Figure 2. The skeletal hand model has 31 DOFs in total, including the translation and the rotation of the wrist.

3.2.2 Surface Hand Model

When all joint positions are determined, the posture of the hand is determined. In order to render the image of a hand, the surface data of the hand skin are needed. The shape of the hand surface must be able to deform according as the skeletal posture. For that purpose, the shape of hand surface is represented by triangle patches, and each vertex of the triangle patch has an attribute that indicates corresponding skeletal link.

4 Hand Pose Estimation

4.1 The Outline of Proposed Method

The estimation of each joint angle is performed by fitting the surface hand model to the voxel model. There can be two approaches for model fitting using 2D observed data and 3D model.

A. 3D model is projected in 2D plane, and the model fitting is performed in 2D plane(Conventional approach).

B. 3D shape is reconstructed by combining 2D observed data, and the model fitting is performed in 3D space(Our approach).

Our method belongs to the latter approach, which directly deals with the 3D deformation of the shape of the model. Therefore, the model fitting in our method can be performed based on a simpler algorithm than those of conventional ones.

The voxel model represents the area where a hand occupies in the voxel space. The surface hand model also represents the position where the hand exists in terms of vertex coordinates of triangle patches. When the surface hand model is completely included in the voxel model, it is considered that 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 i-th joint in axis-k), the hand posture can be represented as $P = \{\mathbf{a}_i|0 < i < r\}$ (r is number of all joints). In this posture, 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 vertex coordinate, q is number of vertices). Each $\mathbf{p}(m)$ is determined by P. Then V is defined as the occupied area of a voxel model. The hand pose estimation is now the process to find 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 make skeletal hand model approach to voxel model is generated for each point included in Out. This process is iteratively performed with changing the joint angles gradually by the generated force vector.

4.2 Detail of Estimation Algorithm

The detail of the estimation is described as follows.

Step1: Silhouette images are created using captured images by the multi-viewpoint camera system.

Step2: The voxel model is created from these silhouette images.

Step3: The skeletal hand model representing the hand pose is compared with the voxel model representing the observed hand shape. The vertices of the triangle patches located at outside of the voxel model are focused on.

Step4: The force f that has the direction to a joint axis is generated about the focused vertex as shown in Figure 3. The generated force f is then converted to the torque around the joint, which is termed as t. t is summed up and the total torque T is determined.

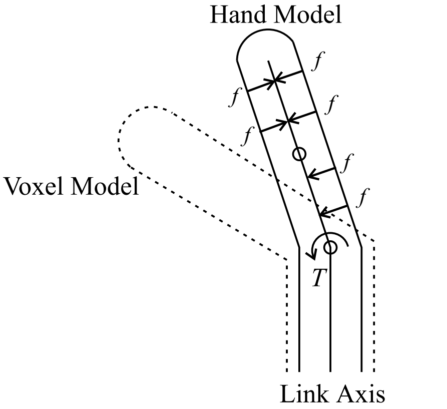


Figure 3: Scheme of model fitting

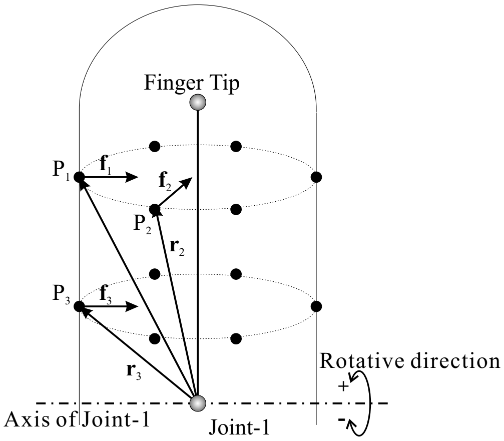


Figure 4: Generation of torque

Step5: The joint angle is changed by $\Delta\alpha$ according as the direction of the torque.

Step6: The joint positions are recalculated from new joint angle and the coordinates of the vertices in the surface hand model are updated.

Step7: The evaluation function is calculated. If the result of evaluation is below a threshold, the estimation finishes. Otherwise, the next process goes back to Step3.

How to determine the rotative direction of each joint in Step4 is explained using Figure 4. It is assumed that vertices P_1, P_2 and P_3 are now located in the outside of the voxel model. It is given beforehand that these vertices are related to rotation of Joint-1. At this time, perpendicular forces to a joint axis are given to each vertex described as $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$. Moreover, the vector from the position of a related joint

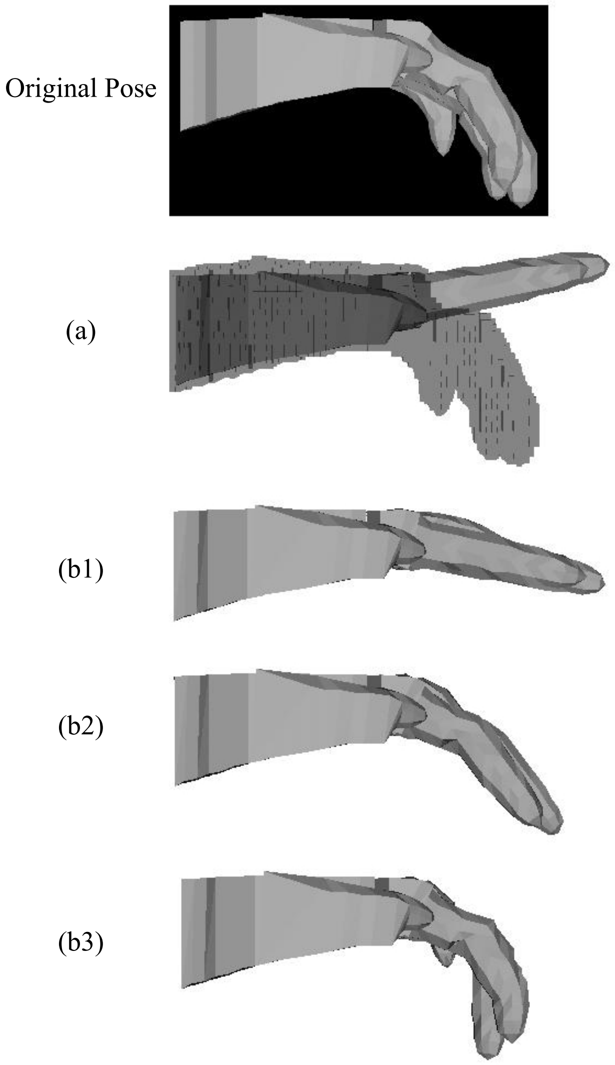


Figure 5: Convergence process
octree level = 8 (minimum octant = 2mm cube)

(Joint-1) to these vertices are defined $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$. Then the torque $\mathbf{t}_i$ in these vertices are calculated by the distance vector $\mathbf{r}_i$ and the force vector $\mathbf{f}_i$. Each of these vertices torque are totaled up to $\mathbf{T}$ that is the rotation torque of Joint-1. The rotative direction about the axis of Joint-1 is determined using the direction of torque $\mathbf{T}$. In the case of Figure 4, Joint-1 is rotated $+\Delta\alpha$ degrees.

In Step7, the calculation of the convergence ratio is performed for each joint. The convergence ratio is defined as follows

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

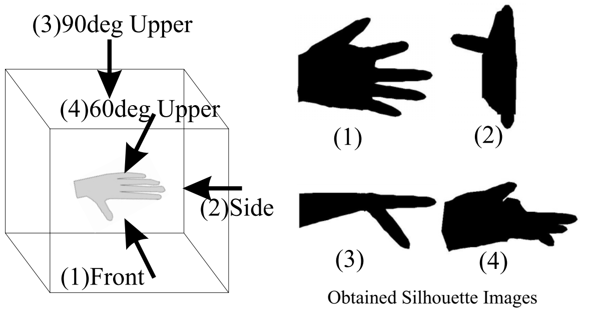
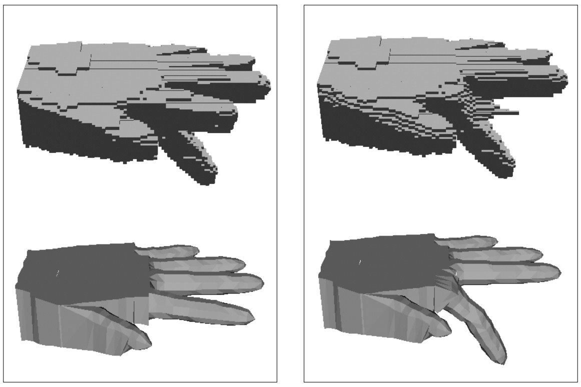


Figure 6: View points and obtained silhouette images



(a) three cameras
Viewpoint (1),(2),(3)

(b) four cameras
Viewpoint (1),(2),(3),(4)

Figure 7: Effect of camera positions
(upper:voxel model[level=8] lower:estimated pose)

where *rate* is convergence ratio, *in_vertex* is the number of V that locate inside the voxel model, *all_vertex* is the number of V. V are the vertices that are related to the rotation of the focused joint. When all 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, when an estimated angle exceeds the movable range of the joint or the direction of torque begins vibration, the angle estimation of a finger is finished.

5 Hand Pose Estimation Using Simulator

5.1 Result of Estimation

We generated various kinds of hand poses using a hand pose simulator, and estimated the hand pose. An example of estimated results is shown in Figure 5. In figure, (a) shows a voxel model together with the initial hand model, and (b1) $\sim$ (b3) show the convergence processes of the surface hand model to the

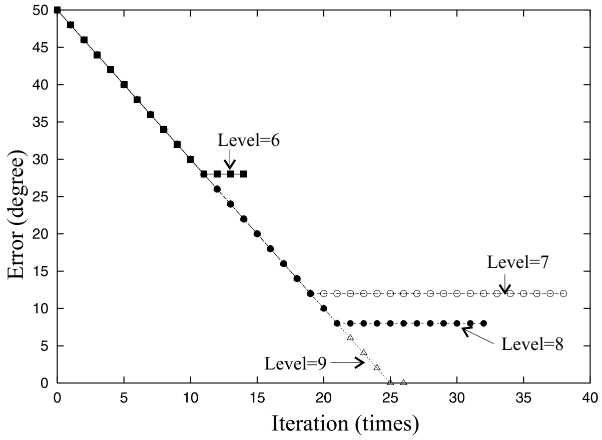


Figure 8: Estimation error of MP-2

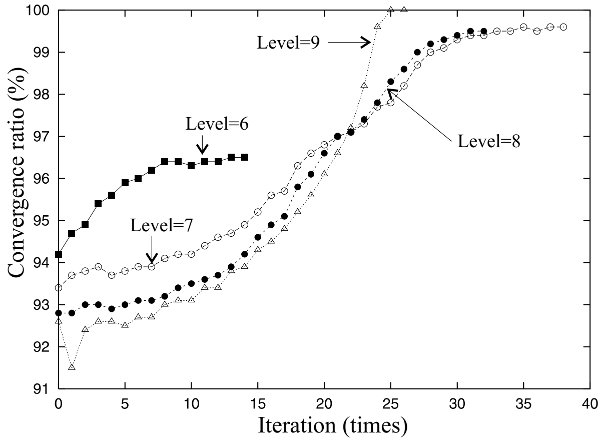


Figure 9: Convergence ratio

voxel model. (b3) shows the final estimated pose of the hand model, which is completely included in the voxel model.

5.2 Effect of Camera Position

The camera position and number affect the accuracy of the voxel model greatly. A experiment was conducted to confirm the effect of the camera position. The estimated hand pose is as follows.

- The MP joint of the index finger is bent by 50 degrees to the inside.
- Other fingers are not bent.

The result of voxel generation using three cameras (No.1, No.2 and No.3) is shown in Figure 6(a). Since a wrong index finger due to the occlusion is generated, the estimated pose of the index finger is wrong. After adding camera No.4, the wrong index finger in the reconstructed voxel model almost disappeared and the estimated pose of the index finger is both correct as shown in Figure 6(b). This experiment indicates that the number and configuration of the cameras are very important in order to generate a correct voxel model of a hand.

Table 1: Result of estimation

	MP-1	MP-2	PIP	Speed
Target pose	0	50	0	
Level 6	-3	22	0	268
Level 7	0	38	32	709
Level 8	-1	42	18	1183
Level 9	0	50	0	3199

unit : degree, msec
CPU : PentiumIII 1GHz Dual

5.3 Effect of Octree Level

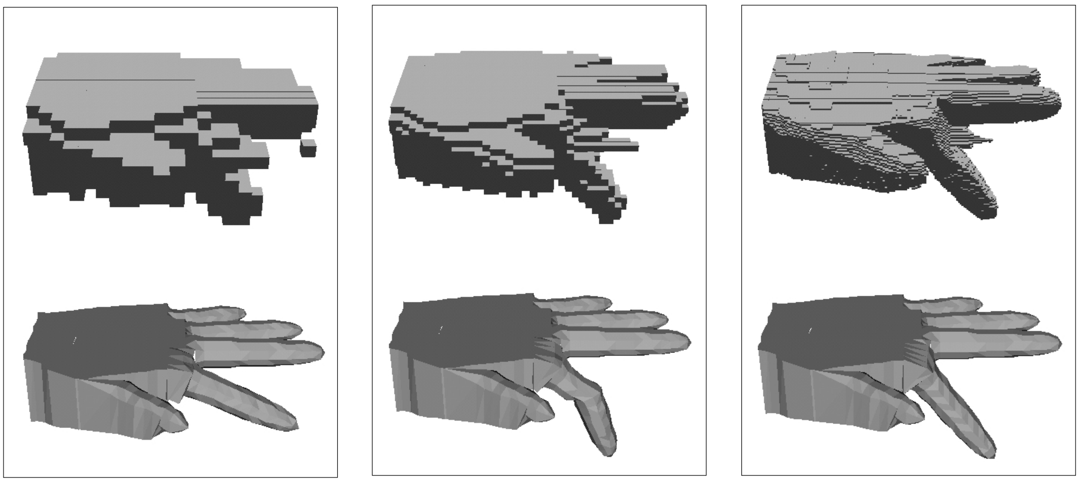
The accuracy of the estimation depends on the maximum level of the octree. The minimum octant of the voxel model is 8mm cube at level 6, 4mm cube at level 7, 2mm cube at level 8 and 1mm cube at level 9 respectively. A quantitative experiment was conducted to confirm the relationship between the octree level and the accuracy of the estimation. The estimated hand pose is same to 5.2.

Figure 8 shows the experimental results of the estimation error in the simulation with octree level 6-9. The estimation errors after every iterations are shown in Figure 8. The estimation error is defined as follows

$$error = |e\_ang - t\_ang|$$

where e_ang is the estimated angle, and t_ang is the true angle. Figure 8 indicates that the accuracy in the estimation becomes higher according as the octree level being higher. Figure 9 shows the profile of the convergence. The results of estimation are shown in Figure 10 and Table 1. In Table 1, MP-1 is an abducent angle of the index, MP-2 is a flexural angle of the 3rd joint of the index, PIP is a flexural angle of the 2nd joint of the index, Speed is the processing time. Except octree level 9, it was estimated that the adjacent joint was also crooked.

These figures and table indicate that the level of the octree gives influences to the accuracy in the estimation and the convergence speed. When the octree level becomes high, the accuracy becomes high while the convergence becomes slow. Therefore, the level of the octree should be determined according as the desired accuracy and the processing time.



(a) Octree level = 6
(Minimum octant= 8mm cube)

(b) Octree level = 7
(Minimum octant= 4mm cube)

(c) Octree level = 9
(Minimum octant= 1mm cube)

Figure 10: Effect of octree level (upper:voxel model, lower:estimated pose)

6 Conclusion

In this paper we proposed a novel hand pose estimation method, which was designed for interfaces of 3D free-form input systems. In the method, a hand is represented by a hand model that consists of the skeletal hand model and the surface hand model. A voxel model is reconstructed from silhouette images of the hand obtained from the multi-viewpoint camera system. The joint angles of the skeletal hand model are estimated by fitting of the surface hand model to the obtained voxel model. In border to confirm the feasibility of the proposed method, we generated various kinds of hand poses using a hand pose simulator, and estimated the hand pose.

As a feature work, we are going to apply the proposed method to a real camera system for vision-based human interface.

References

[1] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang. "Visual Interpretation of Hand Gestures for Human-Computer Interaction : A Review". *IEEE PAMI*, Vol. 19, No. 7, pp. 677–695, 1997.

[2] Akira Utsumi, Jun Ohya, Ryouhei Nakatsu. "Multiple-Hand-Gesture Tracking using Multiple Cameras". In *Proc. of International Conference on Computer Vison and Pattern Recognition*, pp. 473–478, 1999.

[3] C. Maggioni, B. Kämmerer. "Gesture Computer — History, Design and Applications". In *Computer Vision for Human-Machine Interaction.* Cambridge University Press, 1998.

[4] Nobutaka Shimada, Yoshiaki Shirai, and Yoshinori Kuno. "3-D Pose Estimation and Model Refinement of An Articulated Object from A Monocular Image Sequence". In *Proc. of The 3rd Conf.on Face and Gesture Recognition*, pp. 268–273, 1998.

[5] Yoshinari Kameda, Michihiko Minoh, and Katsuo Ikeda. "Three Dimensional Pose Estimation of an Articulated Object from its Silhouette Image". In *Proc. of Asian Conference on Computer Vision '93*, pp. 612–615, 1993.

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

[7] Larry Davis, Eugene Borovikov, Ross Culter, David Harwood and Thanarat Horprasert. "Multiperspective Analysis of Human Action". In *In Third Int. Workshop on Cooperative Distributed Vision*, pp. 189–223, 2000.

[8] Richard Szeliski. "Rapid Octree Construction from Image Sequences". *CVGIP:Image Understanding*, Vol. 58, No. 1, pp. 23–32, July 1993.

[9] Yoshihiro Yasumuro, Qian Chen, and Kunihiro Chihara. "Three-dimensional modeling of the human hand with motion constraints". *Image and Vision Computing*, Vol. 17, No. 2, pp. 149–156, 1999.