Incremental Mesh Modeling and Hierarchical Object Recognition using Multiple Range Images

Ryusuke SAGAWA Kei OKADA Satoshi KAGAMI
Masayuki INABA Hirochika INOUE
Information Engineering Course, University of Tokyo
7-3-1, Hongou, Bunkyou-ku, Tokyo, 113-8656 Japan
{sagawa,k-okada,kagami,inaba,inoue}@jsk.t.u-tokyo.ac.jp

Abstract

This paper describes a vision system which recognizes 3-D objects in real-time by modeling the shapes of objects and matching the generated models. We have developed the following methods for practically solving important problems of integration such as the estimation of sensor accuracy as well as real-time processing: 1) We reduce the computation of signed-distance, which is necessary to apply the Marching Cubes Algorithm, and select the optimal resolution of models to be generated using Octree, thereby enabling us to generate hierarchical mesh models in real-time. 2) We apply spin-image matching, selecting the resolution of generated models and the coarse-to-fine algorithm; consequently, we are able to efficiently match multiple objects of different sizes.

1 Introduction

In this paper, we propose an object-recognition system in 3-D space. Our approach is as follows: 1) modeling the shape of objects by observing them from multiple viewpoints, and 2) selecting matches of the generated models.

In previous work on real-time recognition, researchers solved comparatively simple problems by processing a single image or by recognizing simple-shape objects with parametric models [1, 2]. On the other hand, many studies focused on modeling the shape of an object by using range images obtained by the use of computer graphics . They used a mesh model to reconstruct a complex shape. [3, 4, 5, 6]. However, they did not take real-time processing into account. Sumi et al.[7] and Wheeler[4] used matching of their model generated from range images to recognize objects. The resolution of their model was fixed because the size of the objects was almost identical in their studies.

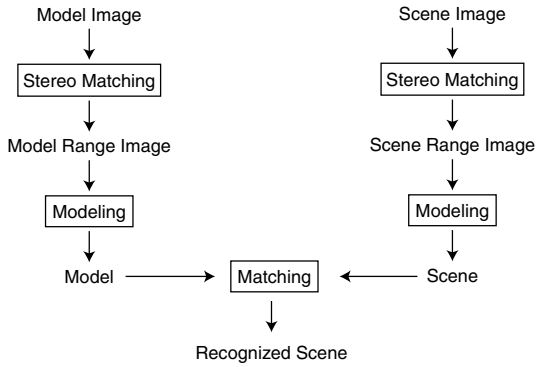

Figure 1: Block diagram of this system

When we develop an object-recognition system in 3-D space, it is necessary to solve the problems which are generated in integrating range images into a model, problems which include the 3-D model resolution control and real-time processing. For real-time recognition of 3-D objects, we propose the following procedures: 1) We model an object using a hierarchical mesh model. The model is generated in real-time by controlling the resolution and the cost of computation. 2) We focus on rough localization of an object to be recognized in real-time. We efficiently select matching multiple objects of different sizes by selecting those that match the resolution of the generated hierarchical model.

Figure 1 shows the system we propose. First, range images are acquired by a range sensor. Second, the model of an object is reconstructed using range images. Finally, we use matching between the scene and the model to recognize the object.

In this paper, Section 2 and Section 3 describes the method of generating a mesh model using ranges

images and the real-time algorithm with a hierarchical mesh model. Section 4, Section 5, describes the method for matching mesh models and its improvement using a hierarchical model. Finally, Section 6 describes the experimental use of our method in the virtual and real environments.

2 Mesh Model Generation Using Range Images

2.1 Selection of 3-D Modeling Method

Previously, the models which were used in real-time applications like robots were 2-D models (image template, line drawing, etc.) and parametric models (generalized cylinder[1], super-quadrics[2], etc). However, 2-D models are insufficient for representation of 3-D information. And, while parametric models can reduce required memory and computation and can be sufficiently robust to recognize objects despite noise, they do not satisfactorily represent arbitrary complexity, and are difficult to generate from sensor data such as range images. Furthermore, because parametric models are composed of only a few primitives, it is difficult to recognize them in scenes where objects are cluttered and partially occluded,.

Fortunately, because of the recent increase in computing power, we can use mesh models of large data to represent 3-D shapes. Mesh models are composed of many primitives (vertices, edges and normals) which have few parameters. The advantages of using mesh models in modeling process are as follows: 1) Adjusting the resolution of a mesh model enables us to represent the shape of arbitrary complexity. 2) Since a mesh model can be divided into partial mesh models, we can take matching using partial ones. Therefore, our method uses mesh models for modeling and matching.

2.2 Volumetric Representation

The resolution of a mesh model generated by a range image varies according to the distance from the viewpoint. In order to merge range images from multiple viewpoints, we use the representation which is independent of the viewpoint. We divide 3-D space into voxels and limit the existence of mesh vertices to only the voxel edges (Figure 2). So, the mesh resolution is independent of the viewpoint and the model is a volumetric representation.

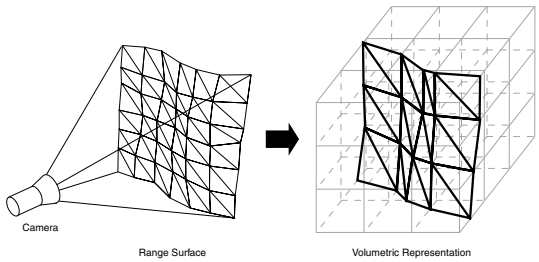

Figure 2: Projective representation to volumetric representation

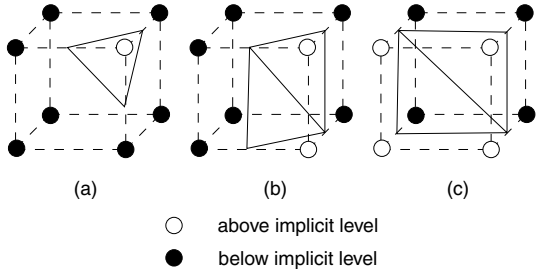

Figure 3: Generate surface mesh by Marching Cubes Algorithm

2.3 Marching Cubes Algorithm

The Marching Cubes Algorithm [8] proposed by Lorensen and Cline is the method for generating volumetric mesh models. Input for this method is a set of voxels, and vertices of each voxel are given scalar values. We denote scalar value of vertices $\boldsymbol{v}_j (j = 1, \ldots, 8)$ of voxel V by $Z(\boldsymbol{v}_j)$. And the input set of voxels is given by

$$S_V = \{V_i | i = 1 \ldots N\}. \tag{1}$$

The meaning of scalar value $Z(\boldsymbol{v}_j)$ is as follows:

$$\begin{array}{ll} Z(\boldsymbol{v}_j) \geq 0 & \text{vertex } \boldsymbol{v}_j \text{ is outside of the object.} \\ Z(\boldsymbol{v}_j) < 0 & \text{vertex } \boldsymbol{v}_j \text{ is inside of the object.} \end{array} \tag{2}$$

According to this, each vertex is in either state, outside of or inside of the object. If the states of adjacent vertices are opposite, the surface of the object intersects these vertices (Figure 3).

2.4 Computation of Signed Distance

Next, we give each vertex of voxels the scalar value $Z(\boldsymbol{v})$. We apply the method proposed by Curless and Levoy[5] and compute $Z(\boldsymbol{v})$ using range images. In

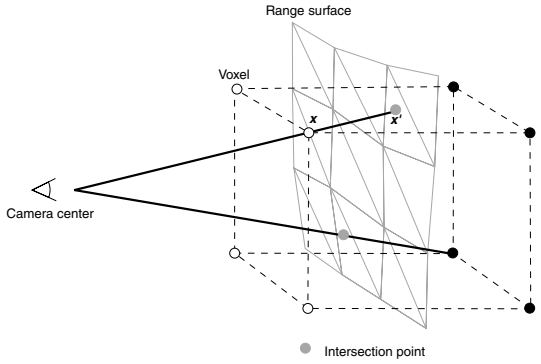

Figure 4: Computation of signed distance

Figure 4, $\boldsymbol{x}'$ is the intersection point of range surface and line which is from the viewpoint of camera to voxel vertex $\boldsymbol{x}$. Then,

$$Z(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}') \cdot \boldsymbol{l} \tag{3}$$

where $\boldsymbol{l}$ is a unit vector parallel to the view vector of camera. If $\boldsymbol{x}$ is closer to the viewpoint than $\boldsymbol{x}'$ is, $Z(\boldsymbol{x})$ is plus; otherwise $Z(\boldsymbol{x})$ is minus. Because the absolute value of $Z(\boldsymbol{x})$ represents the distance from $\boldsymbol{x}$ to range surface, $Z(\boldsymbol{x})$ is called signed distance.

When $Z_i(\boldsymbol{v})(i = 1, \ldots, M)$ is respectively computed for M range images $(i = 1, \ldots, M)$, we compute weighted average of these signed distances; as the result of merging M range images, the final signed distance $V(\boldsymbol{v})$ is given by

$$V(\boldsymbol{v}) = \sum_i w_i(\boldsymbol{v}) Z_i(\boldsymbol{v}) \tag{4}$$

$$w_i(\boldsymbol{v}) = \begin{cases} 1 & |Z_i(\boldsymbol{v})| < T_w \\ \frac{T_w}{Z_i(\boldsymbol{v})} & otherwise \end{cases} \tag{5}$$

where $w_i(\boldsymbol{v})$ is the weighted function and T_w is the appropriate threshold. In this paper, we use $T_w = W_V (W_V$ is width of $V)$.

If range images increase incrementally, $V(\boldsymbol{v})$ can be incrementally updated by

$$\begin{aligned} V_M(\boldsymbol{v}) &= \frac{W_{M-1}(\boldsymbol{v}) V_{M-1}(\boldsymbol{v}) + w_M(\boldsymbol{v}) Z_M(\boldsymbol{v})}{W_{M-1}(\boldsymbol{v}) + w_M(\boldsymbol{v})}, \\ W_M(\boldsymbol{v}) &= W_{M-1}(\boldsymbol{v}) + w_M(\boldsymbol{v}). \end{aligned} \tag{6}$$

2.5 Summary of Modeling Algorithm

In this section, we describe the method of mesh modeling from range images. This method computes signed-distance about all vertices $\boldsymbol{v}$. Therefore, its computation is too costly. Also, the resolution of volumetric representation is given and fixed a priori. We must solve these problems for real-time modeling.

3 Real-time Modeling Algorithm using a Hierarchical Mesh Model

We propose a rapid modeling algorithm for robots; the method contains the following improvements:

1. We reduce the computation by limiting the voxels to be computed signed distance.

2. We enable the representation of models of different resolutions by the use of octree, which represents voxels hierarchically varied in size.

3. We introduce the trade-off between model resolution and computation into the algorithm of computing signed distance, taking into consideration the accuracy of the range sensor.

3.1 Reduction of Computation of Signed Distance

It is necessary to model by use of the Marching Cubes Algorithm that the signed distance is computed on N voxels in voxel set S_V of (1). If there are $A \times A \times A$ voxels in 3-D space, the computation of signed distance is $O(A^3)$. It is much too costly for robot vision in which range images to be processed are given one after another. We reduce the cost by limiting the voxels to be computed signed distance.

Because the voxels actually used for generating surface meshes are those with both plus and minus signed distances, it is necessary to compute signed distance only near the range surface of the objects. We limit the voxels to be computed in the following way.

We find voxels in which the range surface ???is. the voxel set S_V to be computed signed distance is???

$$S_V = \{V_i | n > T_n, S_p = \{\boldsymbol{p}_j | \boldsymbol{p}_j \in V_i, j = 1 \ldots n\}\} \tag{7}$$

where $\boldsymbol{p}_j$ is a vertex of range surface. i.e. S_V is the set of voxels V_i in which the number n of vertices is larger than the threshold T_n.

After limiting by (7), voxels only near range surface are left in S_V. Because the number of voxels in which range surface exists is estimated $O(A^2)$ [4], the computation of signed distance is reduced from $O(A^3)$ to $O(A^2)$.

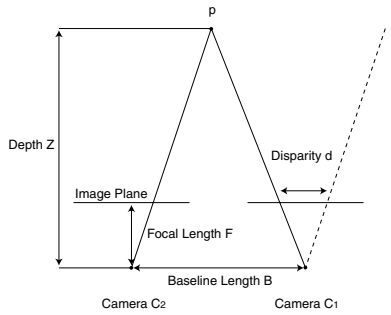

Figure 5: Stereo camera pair

3.2 Hierarchical Volumetric Representation using Octree

We use hierarchically different multi-size voxels to process incrementally by controlling the resolution of volumetric representation according to the shape of the range surface. We use octree [9] to represent hierarchically different multi-size voxels.

We compute signed distance of each size of voxels represented by octree. A highly descriptive model can be generated at high voxel resolution; however, the generated model can be sensitive to noise. It is necessary to control the size of voxels according to the accuracy of range images.

3.3 Considering the Accuracy of Distance by Stereo Matching

Because of various errors, in actual sensing, the element number n of S_p in (7) is smaller than in ideal sensing. If voxel size is too much small, the voxels to compute are also removed by limitation of (7). So, it is necessary to estimate the accuracy of distance measurement and determine the appropriate voxel size.

We use stereo matching for distance measurement because it has advantages in scanning speed and robustness against vibration. We generate a dense disparity map by correlation of window area. Figure 5 shows the geometrical model of stereo matching. If 2 cameras C_1, C_2 observe point p, disparity d is

$$d = \frac{B \cdot F}{Z} \tag{8}$$

where B is baseline distance and F is focal length of cameras.

We consider the decrease of accuracy of distance measurement by the quantization of image. We denote the resolution of disparity by Δd; then the resolution of distance ΔZ is

$$\Delta Z = \frac{\Delta d}{d + \Delta d} \cdot Z. \tag{9}$$

The voxel size should be larger than ΔZ at depth Z to keep voxels to be computed. If the voxel which size is W is at depth Z, the width w of it in the camera image is

$$w = \frac{F}{d_x \cdot Z} \cdot W \tag{10}$$

where d_x is the width of a pixel in the camera image. The number of vertices n in (7) is regarded in proportion to w^2. We denote the threshold $T_n = \alpha^2 w^2$ with a parameter α. Then, to keep voxels which satisfy $W > \Delta Z$, T_n is

$$T_n > \left(\alpha \cdot \frac{F}{d_x} \cdot \frac{\Delta d}{d + \Delta d} \right)^2. \tag{11}$$

If we use actual values, $F = 10.075$(mm) and $d_x = 0.0441$(mm), and assume $\Delta d = 1$ and $d > 9$,

$$T_n > 521.9 \times \alpha^2.$$

We use $\alpha = 0.5 \sim 1.0$. If $\alpha = 0.5$, $T_n = 131$ and if $\alpha = 1.0$, $T_n = 522$.

From another point of view, T_n can be considered a parameter against sensor noise. If $T_n = 100$, signed distances are computed for only voxels which are projected to images larger than 10×10 pixels.

T_n should be larger by the above 2 reasons, but the model resolution result is coarse with large T_n. So, T_n should be decided by the trade-off between the model resolution and robustness against noise.

4 3-D Object Recognition

There has been much research on object recognition by matching 3-D models [4, 10, 11, 12]. They are basically correspondence searches between 2 models and minimization of the distance of the correspondences. We assume that the goal for robots to recognize a object is the detection of the object from a large scene. Then, it is important to estimate roughly the pose of an object before accurate localization of it. We apply the matching method with Spin-Image proposed by Johnson et al [13].

4.1 Spin-Image Matching

Spin-image is the term used to describe a feature of a vertex. In this method, model vertices are mapped

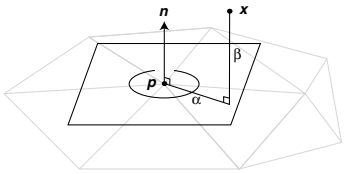

Figure 6: Object-centered coordinate system

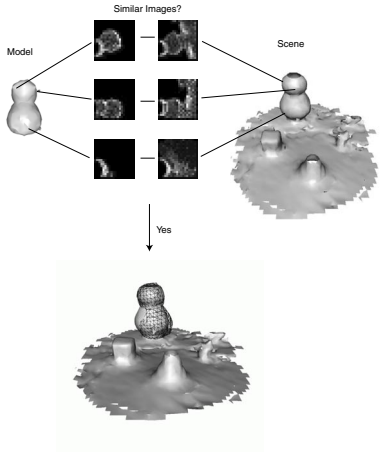

Figure 7: Surface matching concept

to 2-D parameters based on the normal vector of a vertex(Figure 6). A spin-image is a 2-D array in which these vertices are accumulated (Figure 7).

(12) projects vertices to (α, β) which is a coordinate relative to the normal of a vertex. By using the coordinate, the representation of the model shape is independent of its pose.

$$S_O : \boldsymbol{R}^3 \rightarrow \boldsymbol{R}^2$$
$$S_O(\boldsymbol{x}) \rightarrow (\alpha, \beta) = \quad (12)$$
$$(\sqrt{||\boldsymbol{x}-\boldsymbol{p}||-(\boldsymbol{n}\cdot(\boldsymbol{x}-\boldsymbol{p}))^2}, \boldsymbol{n}\cdot(\boldsymbol{x}-\boldsymbol{p}))$$

where $\boldsymbol{p}$ is the position of the base vertex O, and $\boldsymbol{n}$ is the normal vector of vertex O.

4.2 Summary of Spin-Image Matching

The advantages of using spin-images is as follows: 1) Iterative solution is not necessary. 2) Spin-image is smoothly adjustable from local to global representation; thus, it is suitable for matching in a cluttered scene. When we search for objects of various sizes in a large field of view with spin-image matching, it is necessary to select appropriate resolution of a model for computation and accuracy.

5 Matching with Hierarchical Model

We apply spin-image matching to the model generated by our method. We propose the following matching algorithm with a hierarchical model.

1. We select the mesh resolution for matching according to the model to be recognized.
2. We introduce coarse-to-fine strategy and try to match hierarchically from coarse mesh model to fine mesh model.

5.1 Selection of Mesh Resolution

It is important for spin-image matching that the mesh resolution is uniform over the entire model. The method to keep mesh resolution uniform [14] is used in [13]. However, because our mesh model is generated by volumetric representation, its mesh resolution is uniform.

It is desirable that the mesh resolution is automatically selected. But, it is artificially selected by analyzing the generated model. We consider the trade-off between the descriptiveness of spin-image and the cost of matching.

5.2 Recognition Algorithm with Multiple Resolution Mesh Model

The Hierarchical Matching Algorithm we propose matches serially from coarse model to fine model according to the coarse-to-fine strategy. Our matching algorithm, which recognizes multiple objects with a hierarchical mesh model is as follows:

Algorithm *HierarchicalMatching*
Input: scene mesh S
Input: model mesh list L
1. (* selecting M in coarse-to-fine order *)
2. **for** each model mesh $M \in L$
3. **do** *SpinImageMatching*(S, M)
4. remove vertices from S which are matched with M

When the numbers of model and scene vertices are A and B, the computation of spin-image matching is $O(AB)$. As an example, we consider the task of location an object on a table in a scene. The volume of the scene, table and object are $(4Aa)^3$, $(2Aa)^3$ and $(Aa)^3$. When the space between voxel vertices is a, the numbers of these models are $O((4A)^2)$, $O((2A)^2)$ and $O(A^2)$. Also, when the space is $\frac{1}{2}a$, they are $O((8A)^2)$, $O((4A)^2)$, $O((2A)^2)$. We assume that for modeling the table space a is necessary, and for modeling the

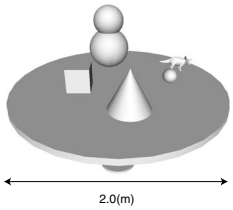

Figure 8: Virtual model of objects on a table

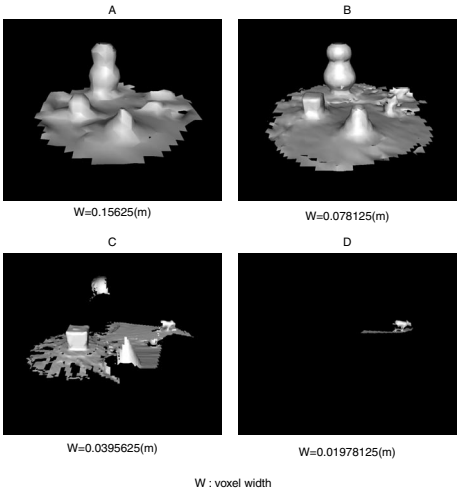

Figure 9: Generated mesh models by different voxel size

object space $\frac{1}{2}a$ is necessary as well. Our method finds the table in the scene with modeling of space a first. And next it finds the object on the table with modeling of space $\frac{1}{2}a$. The computation is

$$O((4A)^2 \cdot (2A)^2 + (2 \cdot 2A)^2 \cdot (2A)^2) = O(128A^4). \quad (13)$$

On the other hand, if we try to find the object from the scene immediately, modeling of space $\frac{1}{2}a$ is necessary. The computation is

$$O((2 \cdot 4A)^2 \cdot (2A)^2) = O(256A^4). \quad (14)$$

Accordingly, it is to costly to find a small object in a large scene. Coarse-to-fine algorithm is efficient for matching objects.

6 Experiment

First, we describe the experimental results of the use of our algorithm a virtual scene in which range

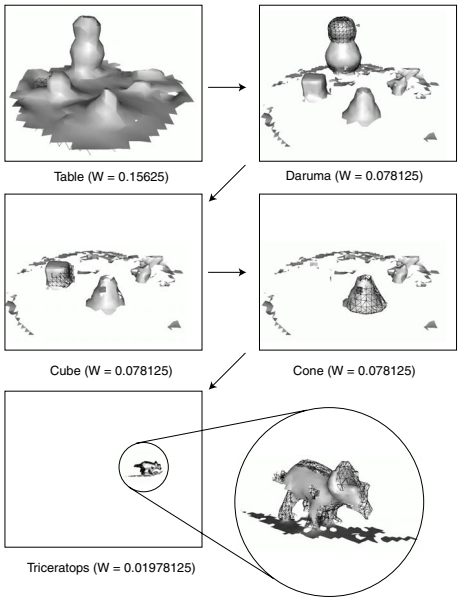

Figure 10: Hierarchical matching process with removing matched vertices

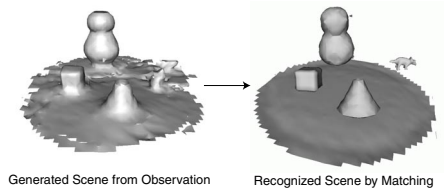

Figure 11: Final matching result

images are artificially generated and no noise exists. We consider the resolution of distance measurement by stereo matching for artificial range image. In the experiment, the scene depicts a table on which are located some objects. (Figure 8).

The results in multiple resolution are showed in Figure 9. It is computed by a PentiumIII 450MHz processor and the average time for each range image is 191(msec).

Our matching algorithm is experimentally used on the model of Figure 8. Models used are table, daruma, cube, cone, triceratops, in that order. The process of hierarchical matching is shown in Figure 10. The final result of matching is shown Figure 11. The time of matching results in 214(sec).

In the next experiment, we use range images generated by stereo matching of multiple cameras. We use 5 cameras shown in Figure 12. A range image is gener-

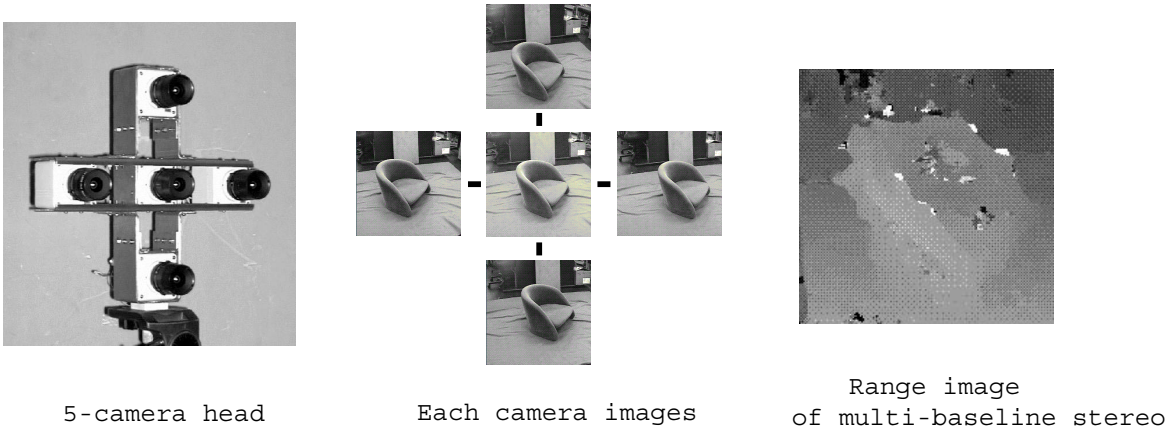

Figure 12: 5-camera head, camera images and a generated range image

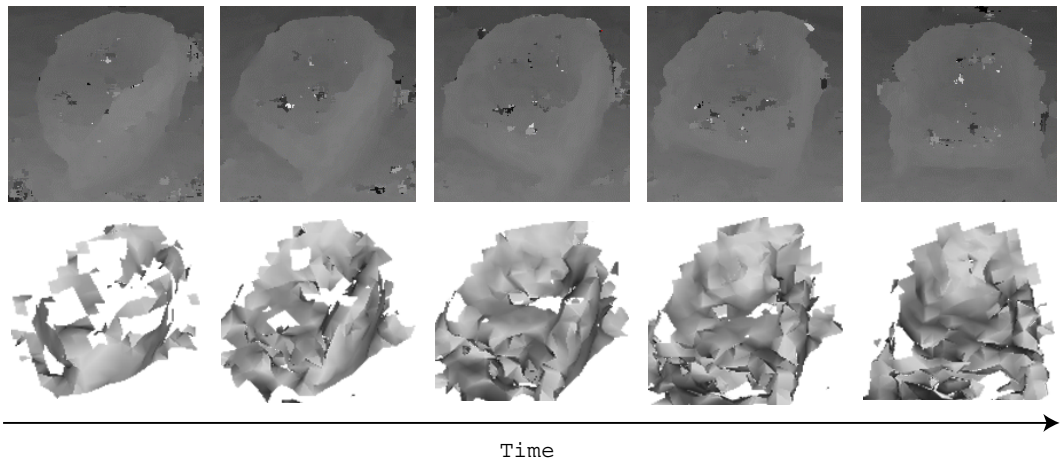

Figure 13: Mesh modeling process of a sofa by 5-camera stereo matching

ated by searching correspondences between the center camera and others using the multi-baseline method[15]. The pose of cameras is measured by a Polhemus sensor. A range image consists of 240×240 pixels. Figure 13 shows the process of modeling a sofa.

The result of matching using generated models is shown in Figure 14. First, we model a sofa and a box. Second, we model the scene with the sofa and box and take matching of each objects. The time needed for matching the two objects is 64 seconds.

In the experiments performed using cameras, the size of objects is restricted according to the abilities of our cameras. Therefore, the advantage of our hierarchical model is not made use of. We can, however, make good use of it if we control the zoom and vergence. Moreover, in the result of recognition (Figure 11 and Figure 14), there are gaps of the pose between the scene and model. For computing an accurate pose, we must apply the ICP method[10, 11], for example.

7 Conclusion

In this paper, we propose a 3-D recognition system which generates the model of an object using range images and then takes matching of generated models. To solve important important integration problems such as the estimation of sensor accuracy and real-time processing, we perform the following procedure: 1) We reduce computation of signed-distance which is necessary to apply the Marching Cubes Algorithm. And also, we select the optimal resolution of

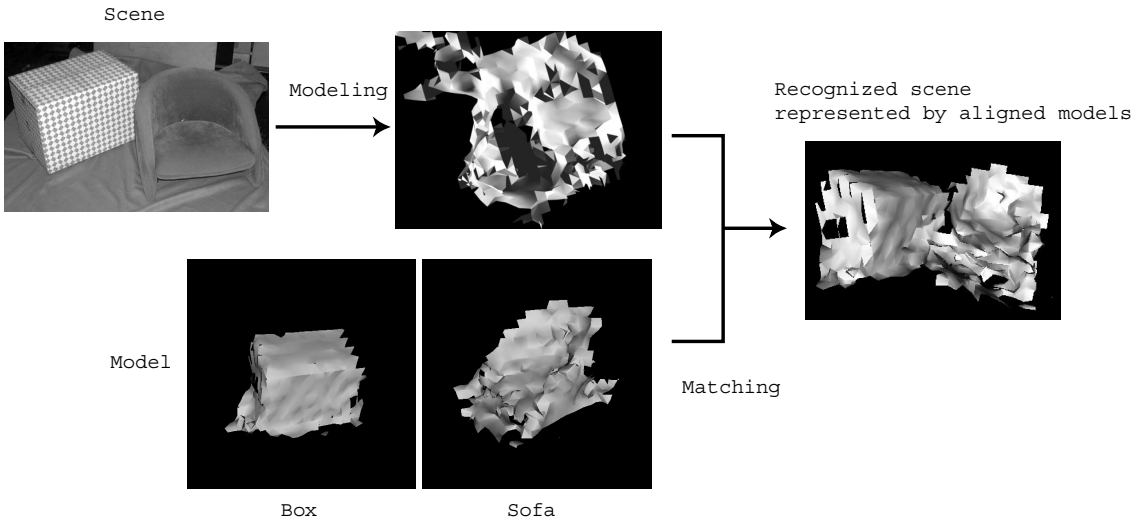

Figure 14: Matching result of a sofa and a box

models to be generated using Octree. Therefore, we generate hierarchical mesh models in real-time. 2) We apply spin-image matching with selecting the resolution of generated models and the coarse-to-fine algorithm. Consequently, we efficiently take matching of multiple objects of different size.

With regard to our future work, we intend to develop a stereo vision system with zooming and vergence control to make use of our hierarchical model. Furthermore, for greater accuracy in modeling, we will develop a method to estimate the camera pose using not only a Polhemus sensor, but also using a visual feedback.

Acknowledgments

This research has been supported by Grant-in-Aid for Research for the Future Program of the Japan Society for the Promotion of Science, "Research on Micro and Soft-Mechanics Integration for Bio-mimetic Machines (JSPS-RFTF96P00801)" project and several grants of Grant-in-Aid for Scientific Research.

References

[1] R.A.Brooks. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 5, No. 2, pp. 140–150, 1983.

[2] N.Raja and A.Jain. Recognizing geons from superquadrics fitted to range data. *Image and Vision Computing*, Vol. 10, No. 3, pp. 179–190, 1992.

[3] A.Hilton, A.J.Stoddart, J.Illingworth, and T.Windeatt. Reliable surface reconstruction from multiple range images. In *Proceedings of the European Conference on Computer Vision*, pp. 117–126, Springer-Verlag, 1996.

[4] Mark D. Wheeler. *Automatic Modeling and Localization for Object Recognition*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1996.

[5] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH'96*, pp. 303–312. ACM, 1996.

[6] S. Vedula, P. Rander, H. Saito, and T. Kanade. Modeling, combining, and rendering dynamic real-world events from image sequences. In *Proceedings of Fourth International Conference on Virtual Systems and Multimedia*, November 1998.

[7] Y.Sumi, Y.Kawai, T.Yoshimi, and F.Tomita. Recoginition of 3d free-form objects using segment-based stereo vision. In *Proc. Int'l Conf. on Computer Vision*, pp. 668–674. IEEE, 1998.

[8] W. Lorensen and H. Cline. Marching cubes: a high resolution 3d surface construction algorithm. In *Proc. SIGGRAPH'87*, pp. 163–170. ACM, 1987.

[9] C.I.Conolly. Cumulative generation of octree models from range data. In *Proc. Intl. Conf. Robotics*, pp. 25–32, March 1984.

[10] P.J.Besl and N.D.Mckay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239–256, 1992.

[11] Z.Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, Vol. 13, No. 2, pp. 119–152, 1994.

[12] C.Chua and R. Jarvis. 3-d free-form surface registration and object recognition. *Int'l Jour. Computer Vision*, Vol. 17, No. 1, pp. 77–99, 1996.

[13] A.E. Johnson and M.Hebert. Efficient multiple model recognition in cluttered 3-d scenes. In *Proc. Computer Vision and Pattern Recognition*, pp. 671–677, 1998.

[14] P.Heckbert and M.Garland. Survey of polygonal surface simplification algorithms. Technical Report CMU-CS-97-TBD, The School of Computer Science, Carnegie Mellon University, 1997.

[15] M.Okutomi and T.Kanade. A multiple-baseline stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, No. 4, pp. 353–363, 1993.