Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3435-3440, Beijing, Oct. 2006.

3D Indoor Environment Modeling by a Mobile Robot with Omnidirectional Stereo and Laser Range Finder

Suguru Ikeda Jun Miura
Department of Mechanical Engineering, Osaka University
{ikeda,jun}@cv.mech.eng.osaka-u.ac.jp

***Abstract*— This paper deals with generation of 3D environment models. The model is expected to be used for location recognition by robots and users. For such a use, very precise models are not necessary. We therefore develop a method of generating 3D environment models relatively simply and fast. We use an omnidirectional stereo as a primary sensor and additionally use a laser range finder. The model is composed of layered contours of free spaces, with textures extracted from images. Results of modeling and application of the model to robot localization are presented.**

I. Introduction

3D environment modeling is one of the active research areas in robotics and computer vision. Most previous works deal with generation of precise 3D models using a large amount of data and elaborate statistical and geometrical estimation techniques. Thrun et al. [1] constructed a multi-planar model from dense range data and image data using an improved EM algorithm. The method was applied to a relatively simple corridor environment surrounded mostly by large vertical planes. Stamos and Allen [2] developed a method of photo-realistic 3D model acquisition from a sequence of 3D range data and that of 2D images. Fleck et al. [3] develop a method of acquiring 3D models by a mobile robot with a laser scanner and a panoramic camera. Sato et al. [4] developed a method of dense 3D reconstruction from a long image sequence with an automatic camera calibration procedure. Nevado et al. [12] have also developed a method of 3D modeling from dense range data. These works mainly focus on generating as precise models as possible to be used for applications such as virtual reality and tele-presence.

This paper deals with generation of 3D indoor environment models with omnidirectional stereo and laser range finder. The models are expected to be used for location recognition by robots and users. Although 2D maps are often used for localization [5], 3D and appearance information will be useful for efficient location recognition. Very precise modeling is, however, not necessary for such a use; rough geometry and appearance would be enough. We thus develop a modeling method of generating such *approximate* models efficiently.

Usual indoor environments in which many objects exist are not formed only by vertical planes. To simplify the modeling process while keeping a certain degree of geometrical information in the model, we represent an environment with a set of layered 2D contours with textures; contours in each layer approximate the shape of 2D free space in a height interval. We use an omnidirectional stereo as a primary sensor and additionally use a laser range finder. Fig. 1 shows the mobile robot used in experiments.

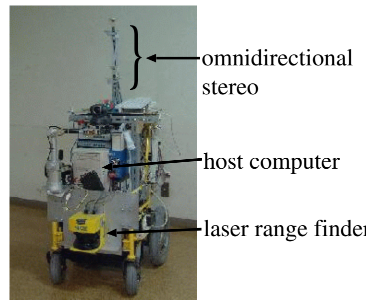

Fig. 1. Our mobile robot.

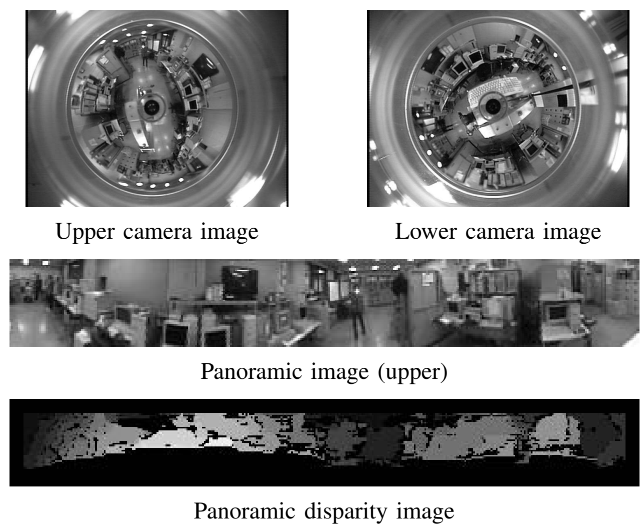

Fig. 2. Result of omnidirectional stereo.

II. Two Sensors

A. Omnidirectional Stereo

The stereo system uses a pair of vertically-aligned omnidirectional cameras (see Fig. 1). The input images are converted to panoramic images, in which epipolar lines become vertical and in parallel; efficient stereo matching algorithms for the conventional stereo configuration can thus be applied. The system can generate the disparity image of 720x100 in size and 80 in disparity range. Fig. 2 shows a pair of omnidirectional images, a panoramic image, and the result of disparity calculation. In the disparity image, larger disparities (nearer points) are drawn in brighter color. Refer to [6] for more details.

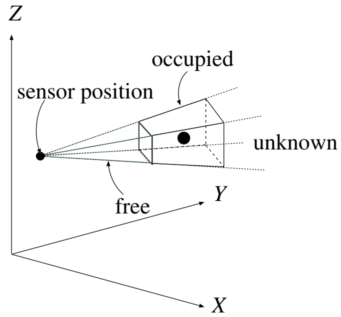

Fig. 3. Voxel classification.

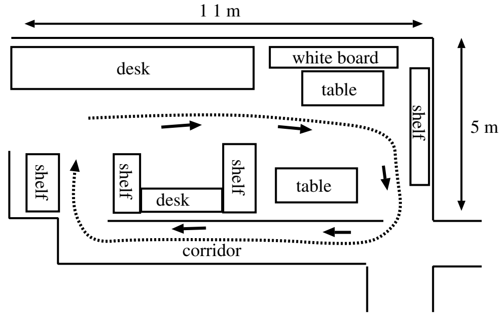

Fig. 4. Experimental environment.

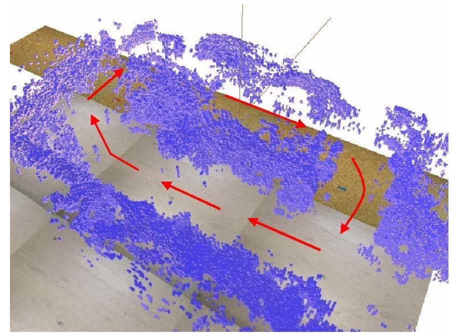

Fig. 5. 3D obstacle map.

B. Laser Range Finder(LRF)

We use a SICK laser range finder (LRF) for obtaining data at lower height. It is set at the front of the robot so that it scans the horizontal plane at the height of $35\,[cm]$ from the floor (see Fig. 1).

III. 3D Obstacle Map Generation

The first step for our 3D environment modeling is to generate a 3D obstacle map. We use a 3D voxel map as the representation of obstacle map. Each voxel is a cube whose edge length is $5\,[cm]$, and holds the probability that an obstacle exist there. We use the data within $3\,[m]$ from the robot position for map generation.

Stereo data usually include not only positional uncertainty but also false data due to stereo matching failure. It is, therefore, necessary to integrate data obtained at various robot positions. This section explains a method of generating 3D obstacle map by temporal integration of stereo data using a probabilistic model of stereo uncertainty.

Rocha et al. [9] developed a method of probabilistic 3D mapping and applied it to a simple environment surrounded mostly by flat walls. Moravec [10] applied an evidence-grid approach to 3D mapping using conventional stereo. We take a similar approach in which we naturally extend our 2D occupancy grid-based mapping method using omnidirectional stereo and LRF[8] to 3D mapping.

A. Determining voxel attributes by one observation

We first determine the attribute of each voxel from one observation. Possible attributes are: *occupied*, *free*, and *unknown*. Fig. 3 shows the attribute determination for a region within one pixel in the panoramic image. The volume around an observed point indicates the uncertainty in the range and direction measurement by the omnidirectional stereo. This volume is labeled as *occupied*. The volume before the occupied is labeled as *free*. The volume behind the occupied is labeled as *unknown* because no information is available in this volume. In the case of stereo, all regions corresponding to the pixels in which any obstacles are not detected (possibly due to the failure of stereo matching) are labeled as *unknown*.

B. Probability update

Let O be the event that an obstacle is detected. O occurs at *occupied* voxels; the inverse event $\overline{O}$ occurs at *free* voxels. For such voxels, the probability is updated as follows. Let E be the event that an obstacle exist, and let $P(E)$ be the probability that an obstacle exist (at a voxel). The new probability map to be obtained by integrating a new observation is given by the conditional probabilities: $P(E|O)$ and $P(E|\overline{O})$. These probabilities are calculated by the Bayes' theorem as follows:

$$P(E|O) = \frac{P(O|E)P(E)}{P(E)P(O|E) + P(\overline{E})P(O|\overline{E})}, \quad (1)$$

$$P(E|\overline{O}) = \frac{P(\overline{O}|E)P(E)}{P(E)P(\overline{O}|E) + P(\overline{E})P(\overline{O}|\overline{E})}, \quad (2)$$

where $P(E)$ is the prior probability and $\overline{E}$ is the proposition that an obstacle does not exist. Among the terms in the above equations, $P(O|E)$ and $P(O|\overline{E})$ are observation models described in [8]; $P(\overline{O}|E) = 1 - P(O|E)$; $P(\overline{O}|\overline{E}) = 1 - P(O|\overline{E})$; $P(\overline{E}) = 1 - P(E)$. Integration for each voxel is performed independently of the others (*the independence assumption*).

C. Map generation

We consider that a voxel with a probability higher than a threshold (currently, 0.8) is occupied by some obstacle. A 3D obstacle map is generated by collecting such voxels (*obstacle voxels*). We guided the robot along the route shown in Fig. 4, and the robot acquired 100 pairs of stereo and LRF data. Each observation location was obtained by our scan-matching method [7]. It took about $390\,[sec]$ to generate a 3D obstacle map shown in Fig. 5. There are 63,147 obstacle voxels in the map.

IV. Generating Layered Contours from 3D Obstacle Map

We use a set of layered contours to model 3D environments. Each layer represents a rough shape of 2D free space in a height interval. We use four layers, three of which from omnidirectional stereo data and one from LRF data, to cope with objects and walls at various heights in usual indoor environments.

To generate a contour fitting to object data in each layer,

We adopt active contour models [11] for determining the contours. All contours in the four layers are simultaneously refined so as to minimize an energy function, which considers the contours' smoothness, fitness to object data, the degree of passing the 2D free space, and consistency between layers. A contour is a set of connected vertical planes and is represented as line segments in 2D. Initial contours are generated from the 2D free space map, which is obtained by our 2D mapping method [8].

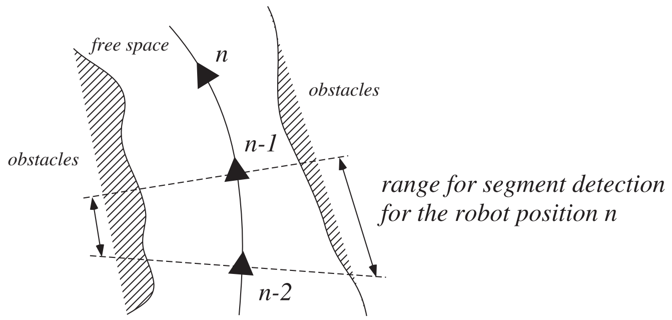

Fig. 6. The range of detecting initial plane segments.

A. Vertical division of the space

The omnidirectional stereo can reliably observe objects whose heights are within the range between $60\,[cm]$ and $160\,[cm]$. Typical objects existing in this height range are tables and chairs, objects on tables such as PC, and shelves. So we divide the height range detected by the stereo into the following three layers:

- *layer 1*: from $60\,[cm]$ to $85\,[cm]$ for tables and chairs.
- *layer 2*: from $85\,[cm]$ to $120\,[cm]$ for objects on tables.
- *layer 3*: from $120\,[cm]$ to $160\,[cm]$ for shelves and other high objects.

Objects lower than $60\,[cm]$ are detected by LRF, which is modeled in *layer 0*.

For layers 1 – 3, we divide the 3D obstacle map into the corresponding three layers and project the obstacle voxels into the 2D map of each layer. Each cell of the 2D map holds the number of obstacle voxels at the corresponding 2D position. We eliminate small clusters of obstacles in the free space.

B. Detection of initial vertical plane segments

As the robot moves, obstacle data on the left and the right side are obtained. We separately detect initial plane segments on each side. We detect line segments by selecting points on the free space boundaries. The detailed steps for initial line segment detection are as follows.

1) Start point selection: At the initial robot position, we determine a start point on the boundary on each side of the robot.

2) Iterative detection of segments: We iteratively extend the line segments when the robot moves by a certain distance. Since the we use range data within $3\,[m]$ from the robot position, the range data $3\,[m]$ behind the robot or further will not change. So the distance is set to $3\,[m]$. Fig. 6 shows a series of robot positions with $3\,[m]$ interval. Suppose the robot is currently at position n. We search the free space boundary at each side for line segments in the range corresponding to positions from $n-2$ to $n-1$ so that the lengths of segments become about $75\,[cm]$. Blue lines in Fig. 7 show the detected initial plane segments. All layers use this same set of segments as initial contours.

C. Refinement of segments using an active contour model

We refine the segments detected above so that they fit better to obstacle data while keeping a certain degree of smoothness and not entering the 2D free map. We also consider consistency between layers in order to generate one 3D geometric map representation.

1) Generation of control points: We use a piecewise-linear model for 2D contours representing vertical planes in 3D. The endpoints of the detected initial segments are used as control points.

2) Energy function: We use the following energy function for each control point:

$$E = \alpha E_{internal} + \beta E_{external} + \gamma E_{between} + \varepsilon E_{free}, \quad (3)$$

where each E_* is a sub-function defined below and α, β, γ, and ε are weights.

$E_{internal}$ is for evaluating the smoothness of a contour and for keeping segment lengths similar. This is defined as the sum of the following three functions:

$$E^1_{internal} = ||\boldsymbol{v}_c - \boldsymbol{v}_p|| + ||\boldsymbol{v}_c - \boldsymbol{v}_n||, \quad (4)$$

$$E^2_{internal} = ||(\boldsymbol{v}_n - \boldsymbol{v}_c) - (\boldsymbol{v}_c - \boldsymbol{v}_p)||, \quad (5)$$

$$E^3_{internal} = |||(\boldsymbol{v}_n - \boldsymbol{v}_c)|| - k| + |||(\boldsymbol{v}_c - \boldsymbol{v}_p)|| - k|, \quad (6)$$

where $\boldsymbol{v}_c$ is the position of the control point currently under consideration, $\boldsymbol{v}_p$ and $\boldsymbol{v}_n$ indicate those of the previous and the next points, respectively, $||\cdot||$ indicates the Euclidean norm and k is a constant.

$E_{external}$ is for evaluating the fitness to obstacle data and defined as:

$$E_{external} = -G * I_{obs}, \quad (7)$$

where I_{obs} is the binary image representation of 2D obstacle map in one layer (one pixel corresponds to $5\,[cm] \times 5\,[cm]$ cell), G is the Gaussian operator and $*$ denotes convolution. σ of G is $4.0\,[pixel]$ in layer 0, and $6.0\,[pixel]$ in the other layers.

$E_{between}$ is for evaluating consistency between layers, and defined using distance D_1 to the contours in neighboring layers as:

$$E_{between} = \begin{cases} -\frac{100}{D_1+1} & (0 \le D_1 < 10) \\ 0 & (10 \le D_1) \end{cases} \quad (8)$$

E_{free} is the penalty for entering the 2D free space, and defined using distance D_2 to the free space boundary as:

$$E_{free} = \begin{cases} D_2 & \text{(inside free space)} \\ 0 & \text{(outside free space)} \end{cases} \quad (9)$$

The summation of E for all control points in all layers is the objective function to be minimized. The minimization steps are as follows (in C-like code):

```
// Sequential minimization
refine_contour(){
  // Minimize energy without E_between in each layer
  apply_acm(α1, β1, γ1, ε1)
  // Minimize energy in all layer
  apply_acm(α2, β2, γ2, ε2)
}
//active contour refinement
apply_acm(α, β, γ, ε){
  while (true) do {
     for i = 3 to 0 do {// Minimize energy in each layer
        for each control point do {
           Move to one of 8-neighbors
           which minimizes the energy of the point; }}
     if no point has moved in this round then break;
}}
```

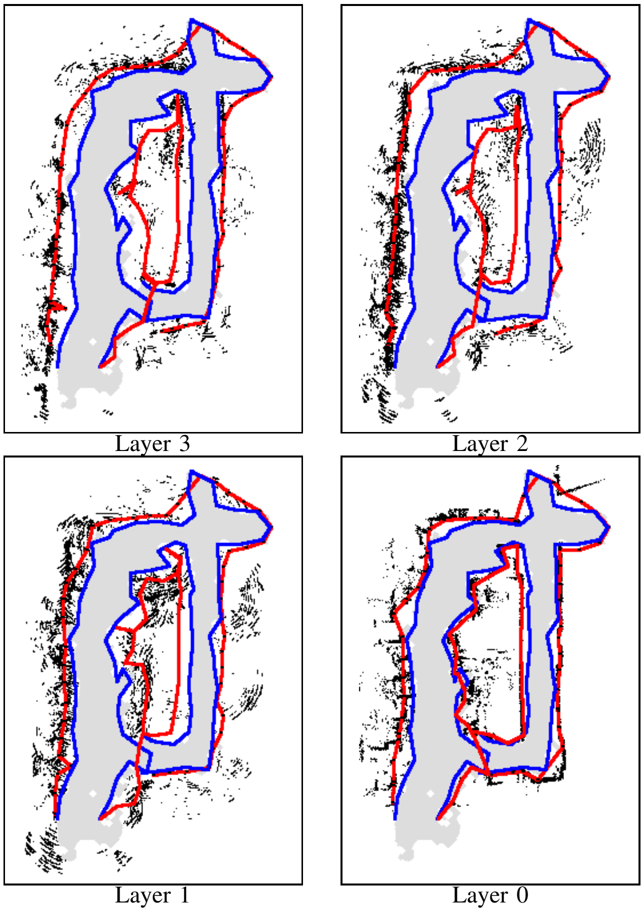

Fig. 7. Initial (blue) and final (red) plan segments. Black points represent obstacle cells in each layer. The gray region indicates the free space.

We currently use the following weights in eq. (3): $(\alpha_1, \beta_1, \gamma_1, \varepsilon_1) = (0.5, 250.0, 0.0, 300.0)$ and $(\alpha_1, \beta_1, \gamma_1, \varepsilon_1) = (0.5, 250.0, 5.0, 300.0)$. Fig. 7 shows the initial and the final plane segments. It is clearly shown that final plane segments fit well to the object data while having sufficient smoothness.

V. Texture Extraction and Mapping

Appearance of obstacles is useful for robot localization and human-robot interface. We therefore extract textures from the images taken by the omnidirectional camera and map them to the constructed plane segments. The mapping between a plane segment and a region in the omnidirectional image is determined by geometry of omnidirectional imaging (see Fig. 8) and the relative position of the segment with respect to the robot position where that image is taken.

We can extract textures for one plane segment from several images taken at various robot positions. In order to get the best textures, we select the image which provides the highest resolution; that is, we select the robot position which maximizes the area of the mapped region of the segment. The extracted textures are stored as images of the size of 256×256 pixels.

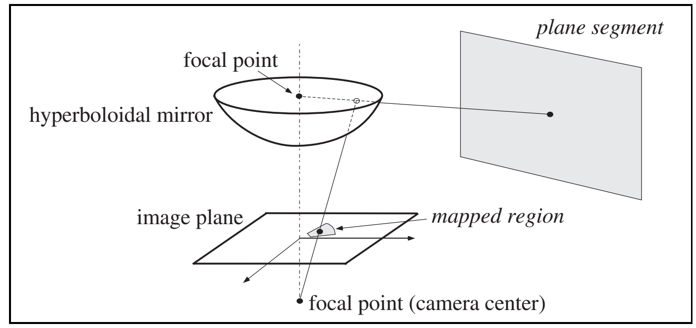

Fig. 8. Mapping a plane onto the omnidirectional image.

Fig. 10. 3D environment model (left) and actual environment (right).

VI. Modeling Results

We have generated a 3D environment model for the environment shown in Fig. 4. The calculation times for generating the initial set of segments and the convergence of active contours for one update were $0.07\,[sec]$ and $7.3\,[sec]$, respectively. The number of updates was 7. The calculation times for texture mapping was $30.0\,[sec]$.

Fig. 9 shows the overview of the acquired 3D model seen from four different viewpoints. It well captures the overall structure of the environment. We manually provided the height of the ceiling and the textures of the ceiling and the floor in advance in generating the final model.

Fig. 10 shows two views of the model observed at some specific positions (left column) and compares them with the real images taken by a digital camera at similar positions (right column). Although discrepancies are found to some extent, it seems sufficient for recognizing the location either by robots and users. The location recognition by robots will be discussed in the next section.

VII. Localization Using the Generated Model

The objective of this research is to develop a method of generating 3D environment models which has the following two properties:

- Simple and thus less costly to generate.
- Contain enough information for location recognition by robots and users.

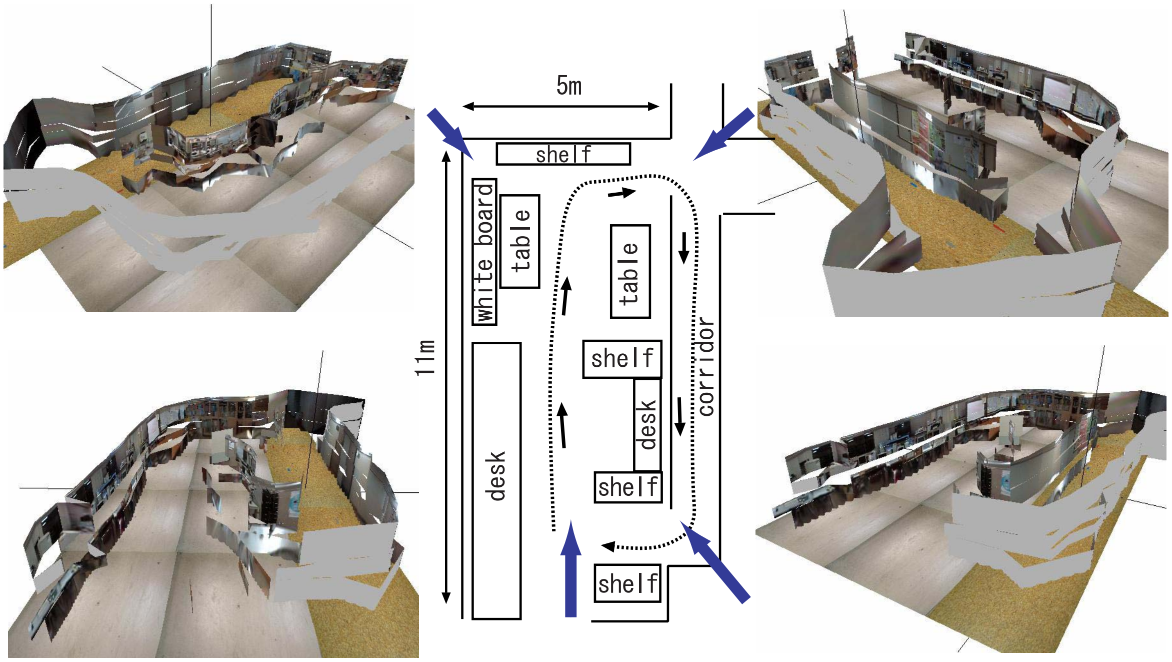

Fig. 9. Overview of the environment model seen from four different viewpoints. Blue arrows indicate their approximate viewing directions.

This section describes a localization method based on the model to show the generated model potentially fulfills the second property.

For localization, we use the observation positions where the robot took data for modeling as candidate points. For each candidate point, we compare the test data with the environment model first using geometric information and then using appearance (i.e., texture) information, and select the best matched point.

The robot generates a local 3D map from several (currently, 10) consecutive data of omnidirectional stereo and LRF, and uses it as the test data for localization.

The distribution of distances in all directions centered at the robot is used as the representation of geometric property of a location. For each candidate point, we extract plane segments in the environment model within $3\,[m]$ distance from the robot and calculate the distribution of distances from the segments. Fig. 11 shows the candidate points and the distribution of distance at several points. We also make the distribution from the local map. We compare two distributions using a simple sum of the absolute difference (SAD) measure with changing orientation (i.e., with shifting the distribution), and select a set of candidate points with associated orientations which have small differences.

We then use texture information for determining the best position. For each remaining candidate point after the geometry matching, we extract the region in the omnidirectional image corresponding to the plane segments using the robot pose (position and orientation) of the candidate point. Fig. 12 shows an example of the mapped region, model texture, and texture of test data. We compare the model and test texture using the SAD measure and determine the position which minimizes the texture difference. Fig. 13 shows a result of localization.

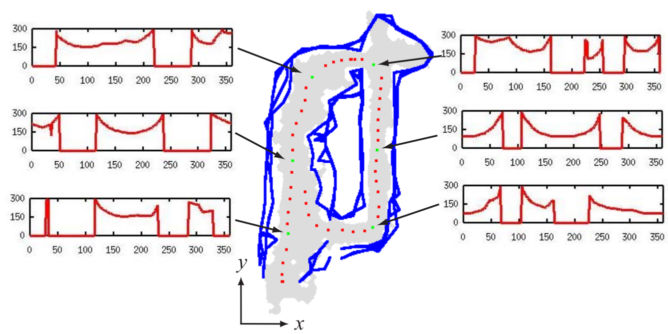

Fig. 11. 50 Candidate points (red) and examples of the distribution of distances (at layer 2) at several candidate points.

We prepared a set of test data taken at 140 positions for localization experiments. Table I shows the results. We tested four cases in which the number of initial candidate points and the number of candidate points kept after the geometry matching. We judge that one of the test data and a candidate point correctly matches if the distance between the position of the data and the candidate point is less than or equal to $1.5\,[m]$. The fourth column of the table indicates the number of test data which have the correct candidate point among the remaining one after the geometry matching. The fifth column indicates the number of test data which have the correct candidate point as the best one. The result shows that our 3D model contains enough information for robot localization. Use of filtering techniques would improve the localization performance.

TABLE I

RESULTS OF LOCALIZATION EXPERIMENTS.

the number of test data	the number of candidate points	the ratio of candidate points after geometry matching	the number of correct geometry matching	the number of correct texture matching	success rate (%)
140	50	5	89	84	60
140	50	10	114	104	74
140	100	10	119	112	80
140	100	20	128	122	87

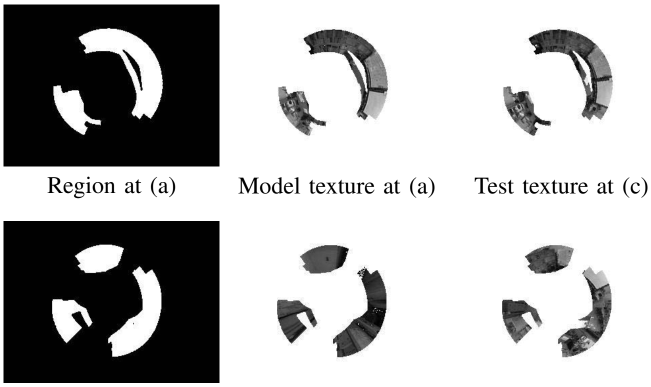

Fig. 12. Mapped regions and textures of model and test data at positions (a), (b), and (c) in Fig. 13.

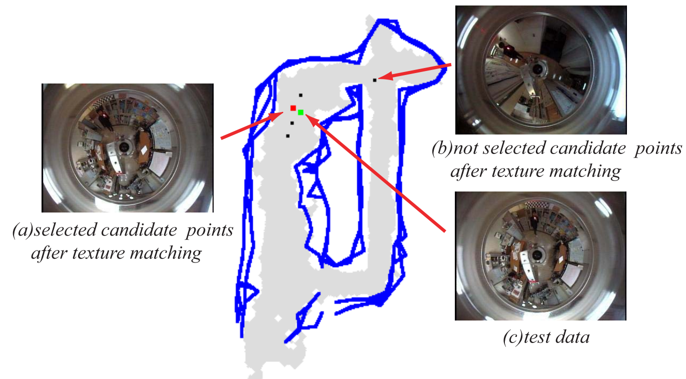

Fig. 13. Result of texture matching. The red point, the green point, and the black points are the selected candidate point, the position of the test data, and the remaining candidate points after geometry matching, respectively.

VIII. CONCLUSIONS AND DISCUSSION

This paper has described a method of generating 3D indoor environment model by mobile robot with omnidirectional stereo and laser range finder. The model is relatively simple to generate but has enough geometric and texture information to be used for location recognition by robots and users. The model is generated by the following three steps: generation of 3D obstacle map by temporal integration of stereo and LRF data, detection of plane segments in four layers using active contour model, and extraction and mapping of textures from images. The resultant models look reasonable even for a complex indoor environment.

The paper has also examined the applicability of the model to robot localization. In localization, we compare a local 3D map with the global model in terms of geometry and texture. By additionally using texture information, the robot can be localized even in the case where there are several geometrically-similar locations.

A future work is to speed up the modeling process. We currently use all data acquired during the robot movement. Since the sensors have large field of view, however, a part of data could be enough for modeling, and reduction of data size will decrease the computation time. We are now investigating the relationship between the size of data and the quality of the generated model. We also plan to test the method to various indoor environment to examine its robustness. Another future work is to use the model for location recognition by users. Especially we expect that the model is used for the interface for communicating location information between the robot and the user.

REFERENCES

[1] S. Thrun et al.: A Real-Time Expectation Maximization Algorithm for Acquiring Multi-Planar Maps of Indoor Environments with Mobile Robots, *IEEE Trans. on Robotics and Automation*, Vol. 20, No. 3, pp. 433–442, 2003.

[2] I. Stamos and P.K. Allen: Geometry and Texture Recovery of Scenes of Large Scale, *Computer Vision and Image Understanding*, Vol. 88, pp. 94–118, 2002.

[3] S. Fleck et al.: Omnidirectional 3D Modeling on a Mobile Robot using Graph Cuts, In *Proceedings of the 2005 IEEE Int. Conf. on Robotics and Automation*, pp. 1760–1766, 2005.

[4] T. Sato et al.: Dense 3-D Reconstruction of an Outdoor Scene by Hundreds-baseline Stereo Using a Hand-held Video Camera, *Int. J. of Computer Vision*, Vol. 47, Nos. 1–3, pp. 119–129, 2002.

[5] S. Thrun: Robotic Mapping: A Survey, Technical Report CMU-CS-02-111, Carnegie Mellon University, 2002.

[6] H. Koyasu, J. Miura, and Y. Shirai. Realtime Omnidirectional Stereo for Obstacle Detection and Tracking in Dynamic Environments. Proc. 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 31-36, 2001.

[7] H. Koyasu, J. Miura, and Y. Shirai. Integrating Multiple Scan Matching Results for Ego-Motion Estimation with Uncertainty, Proc. 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3104-3109, 2004.

[8] J. Miura, Y. Negishi, and Y. Shirai. Mobile Robot Map Generation by Integrating Omnidirectional Stereo and Laser Range Finder. Proc. 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 250-255, 2002.

[9] R. Rocha, J. Dias, A. Carvalho. Exploring Information Theory for Vision-Based Volumetric Mapping In *Proceedings of the 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2409–2414, 2005.

[10] Hans P. Moravec. Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids. CMU-RI-TR-96-34, The Robotics Institute, Carnegie Mellon University, 1996.

[11] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.

[12] M.M. Nevado, J.G.Garcia-Bermejo, and E.Z.Casanova. Obtaining 3D models of indoor environments with a mobile robot by estimating local surface directions. *Robotics and Autonomous Systems*, Vol. 48, pp. 131-143, 2004.