

Estimation of Camera Motion with Feature Flow Model for 3D Environment Modeling by Using Omni-Directional Camera

Ryosuke Kawanishi, Atsushi Yamashita and Toru Kaneko

Abstract—Map information is important for path planning and self-localization when mobile robots accomplish autonomous tasks. In unknown environments, mobile robots should generate an environment map by themselves. Then, we propose a method for 3D environment modeling by a mobile robot. For environmental measurement, we use a single omni-directional camera. We propose a new estimation method of camera motions for improvement in measurement robustness and accuracy. The method takes advantage of a wide field of view of an omni-directional camera. Experimental results showed the effectiveness of the proposed method.

I. INTRODUCTION

Map information is important for path planning and self-localization when mobile robots accomplish autonomous tasks. Therefore, in unknown environment, mobile robots should construct an environment map by themselves.

3D measurement using image data makes it possible to generate map information [1]. However, an image acquired by a conventional camera has a limited field of view [2]. To solve this problem, such cameras with a wide field of view have been invented as a fisheye camera [3], an omni-directional camera [4] [5] and so on. Taking account of installation on a mobile robot, an omni-directional camera is suitable because it can get a surrounding view image at once. Gluckman showed that an omni-directional camera is effective in measurement and recognition in environment [6].

A stereo vision method using two omni-directional cameras is proposed [7]. The measurement accuracy by stereo vision depends on the baseline length. The longer the baseline length is, the better the accuracy is. However, the measurement accuracy of the above method is limited because the baseline length cannot be longer than the robot size. A motion stereo method using stereo pair images which are taken with a single camera at different observation points is equivalent to binocular stereo vision. This method can make the baseline length longer without restriction of the robot size [8]. Therefore, the method can measure with higher accuracy than binocular stereo method.

Measurement accuracy can be expected to be higher by using a sensor fusion method using laser [9], GPS [10] or sonar [11] with an omni-directional camera. However, there are some problems that measurement objects and situations

are limited and that complexity of calibration process for each device increases.

Therefore, we use a single omni-directional camera for environmental measurement. The structure from motion (SFM) is a kind of motion stereo method. The SFM estimates camera motions (the relative relations of camera positions and orientations), and then measures objects in images. We have already proposed a measurement method based on the SFM [12]. Our method extracts and tracks feature points to get corresponding points in an omni-directional image sequence. By using position relations of the corresponding points, the method estimates camera motions and measures its environment. A triangular mesh is generated from measurement data. By texture-mapping, a 3D environment model is constructed.

Estimation of precise camera motion is important for improvement of the accuracy in measurement by the SFM. Then, we propose a method which estimates precise camera motions by taking advantage of an entire field of view of omni-directional camera.

II. OUTLINE

A mobile robot executes 3D measurement and modeling by using an omni-directional camera (Fig. 1). The mobile robot acquires an omni-directional image sequence during its locomotion.

The process of our method is shown in Fig. 2. It extracts and tracks feature points to get corresponding points in an omni-directional image sequence. By using the positions of corresponding points in two images taken at different observation points, we estimate camera motions with a Feature Flow Model we propose. 3D coordinates of object points are calculated by triangulation with camera motions and image coordinates of feature points. In order to decrease accumulated errors, we minimize errors in feature point coordinates and camera motions by bundle adjustment.

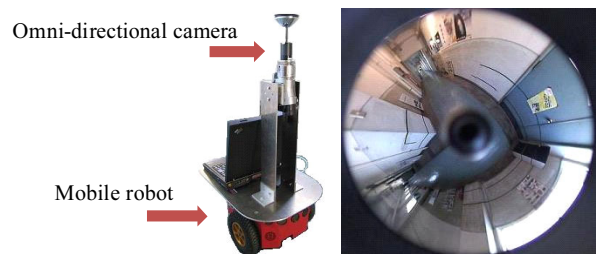


Fig. 1. Mobile Robot and Omni-Directional Image

R. Kawanishi, A. Yamashita, T. Kaneko are with the Department of Mechanical Engineering, Faculty of Engineering, Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu-shi, Shizuoka 432-8561, Japan (e-mail: {f0730034, tayamas, tmtkane}@ipc.shizuoka.ac.jp).

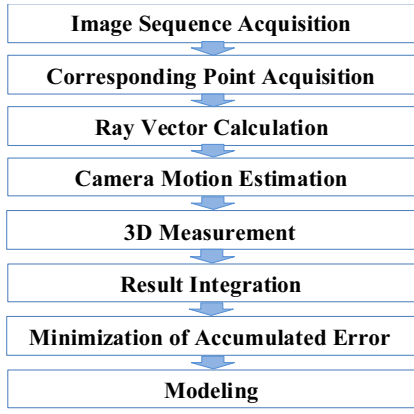


Fig. 2. Process

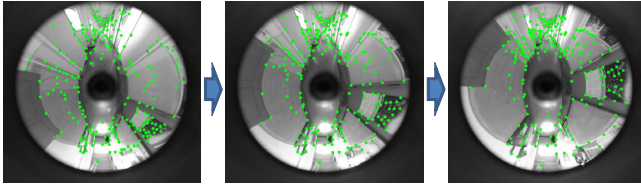


Fig. 3. Feature Extract and Tracking

Finally, the individual measurement data of feature points are integrated as one 3D measurement data.

III. ALGORITHM

A. Corresponding Point Acquisition

For getting correspondent points between images in the omni-directional image sequence, the method extracts feature points in the initial image and then tracks them along the sequence. In our method, we use Lucas Kanade tracker algorithm with image pyramid representation [13] (Fig. 3).

These points are regarded as corresponding points between two images taken before and after the robot movement.

B. Ray Vector Calculation

We define a unit vector originating from the center of projection to an object point in 3D space as a ray vector $\mathbf{r} = [x, y, z]^T$, where T stands for transposition of vector or matrix. An omni-directional camera we use has a hyperboloid mirror in front of the lens of a conventional camera. Therefore, as shown in Fig. 4, ray vector \mathbf{r} is directed from the focus of the hyperboloid mirror to the reflection point of the ray on the mirror surface.

Ray vector \mathbf{r} is calculated from image coordinates $[u, v]^T$ of the feature point using (1), (2).

$$\mathbf{r} = \begin{bmatrix} su \\ sv \\ sf - 2c \end{bmatrix} \quad (1)$$

$$s = \frac{a^2 \left(f\sqrt{a^2 + b^2} + b\sqrt{u^2 + v^2 + f^2} \right)}{a^2 f^2 - b^2(u^2 + v^2)} \quad (2)$$

In these equations, a , b and c are the hyperboloid parameters and f is the image distance which is the distance between a center of a lens and the image plane.

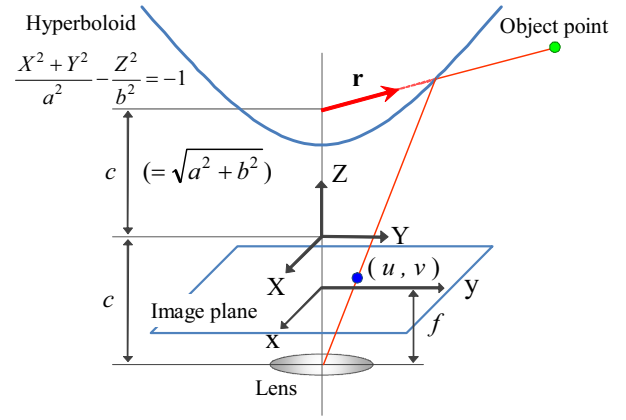


Fig. 4. Calculation of Ray Vector

C. Camera Motion and Feature Point Movement

In the case of using a conventional camera, its field of view is restricted to a narrow area around the optical axis. For this reason, there is not much difference in feature point's movement in images between translation and rotation of the camera (Fig. 5) in several cases. Though these cases do not always happen in general, but they often happen in actual situations. Therefore, it may be difficult to estimate camera motion by analyzing movement of feature points in images.

On the other hand, in the case of using omni-directional camera which has 360 degree field of view, we can specifically discriminate between translation and rotation of the camera by analyzing movement of feature points (Fig. 6).

Then, we propose an estimation method of camera motions by utilizing a difference in feature point movement between camera rotation and translation.

D. Camera Motion Estimation with Feature Flow Model

Our method defines a unit sphere for camera motion estimation. The center of this unit sphere represents the starting points of each ray vector and a camera translation vector. These vectors are unit vectors. Rotation axes of camera motions pass through the center of the unit sphere. A relation between ray vectors \mathbf{r}_1 , \mathbf{r}_2 , camera translation vector

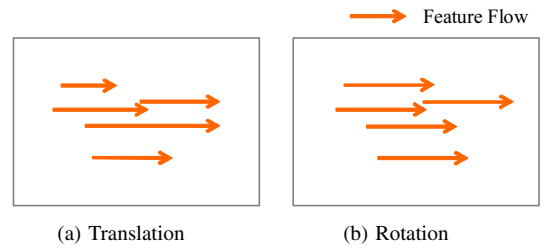


Fig. 5. Feature Flow (Conventional Camera)

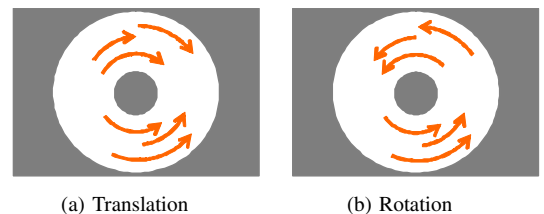


Fig. 6. Feature Flow (Omni-Directional Camera)

\mathbf{t} , camera rotation axis and angle displacements is described below.

In the case of camera translation, a translation vector is in a plane which consists of two ray vectors \mathbf{r}_{t1} , \mathbf{r}_{t2} acquired before and after camera movement. Therefore, a normal vector of this plane given by (3) is orthogonal to a translation vector (Fig. 7 (a)), and this leads to (4).

$$\mathbf{n} = \mathbf{r}_{t1} \times \mathbf{r}_{t2} \quad (3)$$

$$\mathbf{n} \cdot \mathbf{t} = 0 \quad (4)$$

In the case of camera rotation, ray vector's displacements are constant angle displacement in a plane which is orthogonal to a rotation axis. Here, a relation between two ray vectors \mathbf{r}_{r1} and \mathbf{r}_{r2} is described by (5) using a rotation matrix (Fig. 7 (b)).

$$\mathbf{r}_{R1} = \mathbf{R}^{-1} \mathbf{r}_{R2} \quad (5)$$

In the case of both translation and rotation, a relation between two ray vectors \mathbf{r}_1 and \mathbf{r}_2 acquired before and after camera movement can be described by the combination of these displacements. In this unit sphere, these displacements are independent of each other. Therefore, we can deal with this case as well as the case of translation by removing rotation displacements from the combination of two displacements.

Then, we can describe camera motions by following (6).

$$(\mathbf{r}_1 \times \mathbf{R}^{-1} \mathbf{r}_2) \cdot \mathbf{t} = 0 \quad (6)$$

Here, there are two solutions for the translation vector and the rotation matrix which satisfy (6), respectively. In the translation vector, one of the solution is the correct translation vector \mathbf{t} which describes camera motion. The other solution is the vector in the reverse direction of the correct translation vector (reverse translation vector). In the rotation matrix, one of the solution is the correct rotation matrix \mathbf{R} which describes camera motion. The other solution is the rotation matrix which rotates 180 degrees from the correct rotation matrix around the translation vector. So, we should determine which is correct between these two solutions, respectively.

Our method solves the problem by applying two constraint conditions. One of the constraint conditions is that

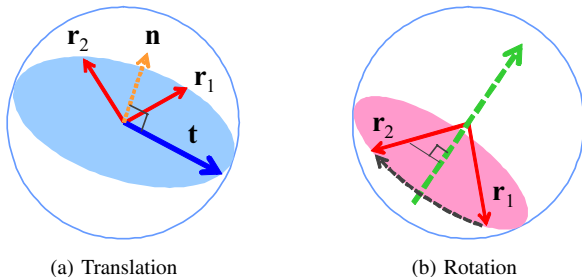


Fig. 7. Feature Flow Model

displacements of ray vectors are in the direction from the end point of the correct translation vector (positive pole) to the end point of the reverse translation vector (negative pole). This constraint condition decides the direction of the correct translation vector. The other constraint condition is that displacements of ray vectors do not cross over the positive pole or the negative pole. This constraint condition decides the correct rotation matrix.

When the rotation matrix and the translation vector satisfy these constraint conditions, ray vectors satisfy (7) geometrically (Fig. 8).

$$\mathbf{q}_1 \cdot \mathbf{q}_2 < 0 \cap \mathbf{q}'_1 \cdot \mathbf{q}'_2 < 0 \quad (7)$$

where

$$\mathbf{q}_1 = \mathbf{R}^{-1} \mathbf{r}_2 - \mathbf{r}_1, \mathbf{q}_2 = \mathbf{t} - \mathbf{r}_1, \mathbf{q}'_1 = -\mathbf{q}_1, \mathbf{q}'_2 = -\mathbf{t} - \mathbf{R}^{-1} \mathbf{r}_2.$$

This equation means that both the angle between vector \mathbf{q}_1 and vector \mathbf{q}_2 and the angle between vector \mathbf{q}'_1 and vector \mathbf{q}'_2 are obtuse.

Our method estimates the rotation matrix and the translation vector which satisfy both (6) and (7) for all the pairs of ray vectors $\mathbf{r}_{1,i}$ and $\mathbf{r}_{2,i}$ by minimizing the amount E given by (8) and (9).

$$e_i = \begin{cases} |(\mathbf{r}_1 \times \mathbf{R}^{-1} \mathbf{r}_2) \cdot \mathbf{t}| & \text{if (7) is satisfied} \\ 1 & \text{else} \end{cases} \quad (8)$$

$$E = \sum_i e_i \quad (9)$$

This method we propose is based on a model which has uniform flow of ray vector's displacements on a unit sphere. So, we call the model of the unit sphere a Feature Flow Model (FFM).

E. 3D Measurement

3D coordinates of an object point which is projected as a feature point in the image are given based on triangulation with two cameras set in the geometrical relation given by rotation matrix \mathbf{R}_m and translation vector \mathbf{t}_m , where m is the number of measurement. We calculate 3D coordinates of $\mathbf{p}_{m,i}$ (i -th feature point) by using \mathbf{R}_m and \mathbf{t}_m (Fig. 9).

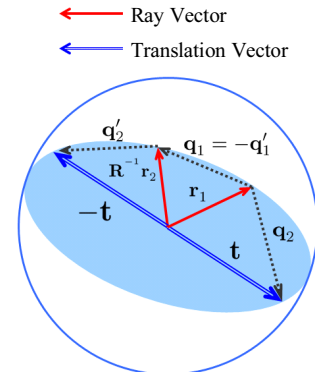


Fig. 8. Constraint Condition

F. Result Qualification

The accuracy of measurement is lower when an object point lies close to the baseline direction or it lies far from the camera. Therefore, the measurement data is a mixture of high and low accuracy. Here, by taking the differentiation of measurement result $\mathbf{p}_{m,i}$ by the image coordinates of the two feature points $[u_{m,i}, v_{m,i}]^T$ and $[u'_{m,i}, v'_{m,i}]^T$ as the estimate of the measurement accuracy, we select measurement results satisfying (10) and (11), where h is a threshold.

$$\mathbf{g}_{m,i} = \left| \frac{\partial \mathbf{p}_{m,i}}{\partial u_{1mi}} \right| + \left| \frac{\partial \mathbf{p}_{m,i}}{\partial v_{1mi}} \right| + \left| \frac{\partial \mathbf{p}_{m,i}}{\partial u_{2mi}} \right| + \left| \frac{\partial \mathbf{p}_{m,i}}{\partial v_{2mi}} \right| \quad (10)$$

$$\|\mathbf{g}_{m,i}\| < h \quad (11)$$

G. Result Integration

By the above procedure, we get individual measurement results and the geometrical relations of observation points by using pairs of the images selected from the image sequence. In order to unite these measurement results, we solve the problem of scale ambiguity among individual measurements by scale matching [14]. However, there is a mismatching of the measurement results because the errors included in feature points are different between each observation point.

Therefore, after scale matching, there are more than two measurement results which show the same object point. We integrate these measurement results into one object point by voting to voxels which are divided in 3D space. The 3D coordinates of the object point are the coordinates of the voxel which has the largest value.

H. Minimization of Accumulated Error

The camera motion estimated in Section 3.4 includes some errors. These measurement errors are accumulated after result integration (Section III–G). For the solution of the problem, Bundle adjustment is known as a minimization method of accumulated errors [15]. To decrease accumulated errors, our method minimizes the errors between the ray vectors calculated by (1) and the vectors whose direction is given by the 3D coordinates of the feature point (reprojected vector). The reprojected vector is calculated by (12) and (13).

$$\mathbf{r}'_{1,m,i} = \mathbf{R}_{m-1} (\mathbf{p}_i - \mathbf{c}_{m-1}) \quad (12)$$

$$\mathbf{r}'_{2,m,i} = \mathbf{R}_m (\mathbf{p}_i - \mathbf{c}_m) \quad (13)$$

Here, \mathbf{c}_m is the camera positions at the observation point m . \mathbf{R}_m is the rotation matrix at the observation point m . We define the accumulated error as follows.

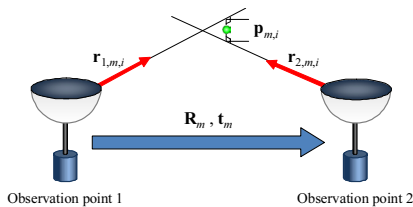


Fig. 9. Calculation of 3D Coordinates

$$\varepsilon_m = \sum_i w_{m,i} (\|\mathbf{r}_{1,m,i} - \mathbf{r}'_{1,m,i}\|^2 + \|\mathbf{r}_{2,m,i} - \mathbf{r}'_{2,m,i}\|^2) \quad (14)$$

where ε_m is the accumulated errors in m -th frame, $w_{m,i}$ is the weight factor for i -th feature point, $\mathbf{r}_{m,i}$ is the ray vector, $\mathbf{r}'_{m,i}$ is the reprojected vector. $w_{m,i}$ is The weight factor which evaluates the measurement error by using the norm of the vector $\mathbf{g}_{m,i}$ calculated by (10). The weight factor $w_{m,i}$ is calculated by (15).

$$w_{m,i} = \frac{1}{\|\mathbf{g}_{m,i}\|} \quad (15)$$

I. Modeling

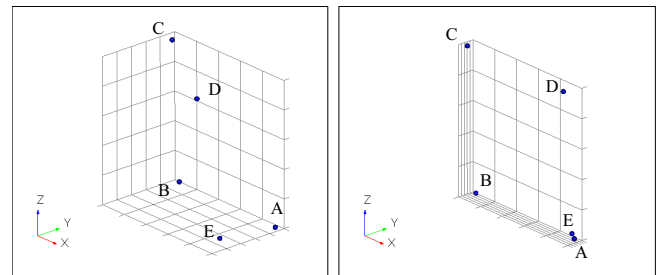
A triangular mesh is generated from integrated measurement data by using 3D Delaunay triangulation. But, the Delaunay triangulation generates a triangular mesh which contradicts a physical shape because the triangular mesh does not consider the shape of the measurement object. Therefore, we apply the triangular optimization method [16] to the triangular mesh. The method adapts the triangular mesh to the physical shape by detecting a texture distortion. Here, generated triangular mesh includes patches which are blind spots of the camera, for example, the ceiling and the floor. Therefore, we remove these patches by using estimated camera poses. By texture mapping to the triangular mesh, a 3D environment model is constructed.

IV. EXPERIMENT

We evaluated the accuracy of camera motion estimation by using an omni-directional camera equipped on a manipulator. We regard a manipulator movement as the true value of camera motion. We made the manipulator trace a square. We estimated camera motion 4 times between each apex of the square. The end position of the camera is same as the starting position. The length of the square side is 350 millimeters.

In this experiment, we compared FFM with the eight-point algorithm. To show the profitability of our method, we did not remove outliers included in corresponding points we got in Section III–A. Figure 10 shows the result of camera motion estimation. Camera positions A, B, C, D and E correspond to apexes of the square which is traced by the manipulator.

In the result of the eight-point algorithm (Fig. 10 (a)), the end position E is far from the starting position A. In the result by a FFM (Fig. 10 (b)), the end position E is close to the starting position (Table I), where we define the distance



(a) Eight-Point Algorithm

(b) FFM

Fig. 10. Result of Camera Motion Estimation

TABLE I
RESULT OF CAMERA MOTION ESTIMATION 1

	Eight-Point Algorithm			FFM		
	X	Y	Z	X	Y	Z
A	0	0	0	0	0	0
B	-0.997	0.075	-0.030	-0.998	0.063	-0.030
C	-1.039	0.053	0.932	-1.024	0.018	0.969
D	-0.140	-0.527	0.981	-0.039	-0.054	0.997
E	-0.049	-0.419	0.028	-0.016	-0.005	0.030
A-E Distance	0.422			0.035		

TABLE II
RESULT OF CAMERA MOTION ESTIMATION 2

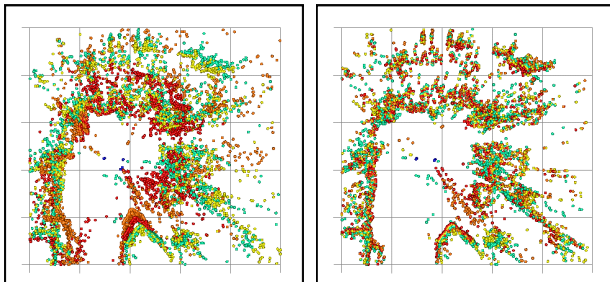
	Eight-Point Algorithm	FFM
θ_1 [deg]	89.3	90.4
θ_2 [deg]	88.7	89.7
θ_3 [deg]	91.6	90.5
θ_4 [deg]	93.2	89.4
Average [deg]	90.7	90.0
Variance [deg ²]	3.23	0.187
Standard Variance [deg]	1.80	0.432
Maximum Error [deg]	3.2	0.5

between A and B is 1 because scale information is unknown in our method. In addition, the result by a FFM has higher accuracy of camera movement direction than the result of the eight-point algorithm (Table II).

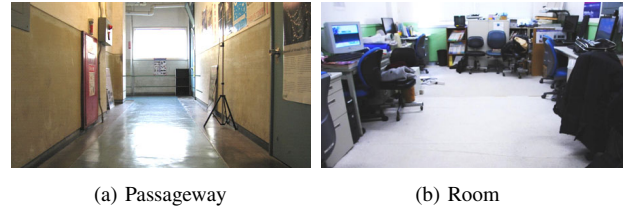
Fig. 11 shows the measurement result around the camera. The blue marks in this result show the trajectory of the camera. The green, yellow, orange and red marks show each measurement data at 4 observation points. The measurement result by a FFM has smaller variance between each measurement result than the result of the eight-point algorithm. These results show that our proposed method by a FFM can estimate with higher accuracy than the eight-point algorithm.

We construct a model of an actual environment by measurement method based on a FFM by using a mobile robot which equips an omni-directional camera. In the experiment we measured two environments (Fig. 12 (a) a passageway including an L-shape corner, (b) a room). We acquired an image sequence (10 fps) of these environments by using a mobile robot. The robot ran with 100 mm/s. The image size is 1920×1080 pixels.

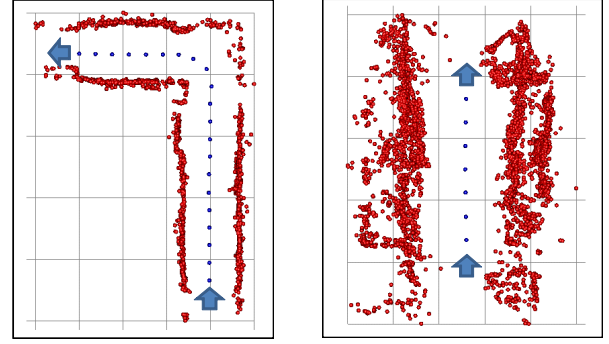
Fig. 13 shows the top view of combined measurement result of the passageway and the room. The blue marks in this result show the trajectory of the robot. The red marks in this result show the measurement data. These results show that our method can measure the shape of the passageway



(a) Eight-Point Algorithm (b) FFM
Fig. 11. Measurement Result of Camera Motion Estimation



(a) Passageway (b) Room
Fig. 12. Experiment Environment



(a) passageway (b) room
Fig. 13. Integrated Measurement Data

and the room.

Fig. 14 and 16 show the 3D environment models of the passageway and the room. Figure 15 and 17 show other views of these 3D environment models. These results show that our proposed method constructs models which have the shape of measurement object.

V. CONCLUSION

In this paper, we propose a method which estimates camera motions precisely by taking advantage of the wide field of view of an omni-directional camera. Experimental results show the effectiveness of our proposed method.

As future works, we should make the following improvements. For measurement accuracy, the baseline length should be optimized by considering the measurement result at each observation point. For appearance of the environment model, we should estimate the shape of measurement objects in the environment from the relation between the measurement result and the texture appearance at each observation point. Moreover, the present experimental environments are not very large. Therefore, we should experiment in large environments, such as outdoors. Further, we should compare other methods which use SIFT[17] feature, SURF[18] feature and so on.



Fig. 14. Environment Model of the Passageway



(a) Door



(b) Panel



(c) Fire Hydrant

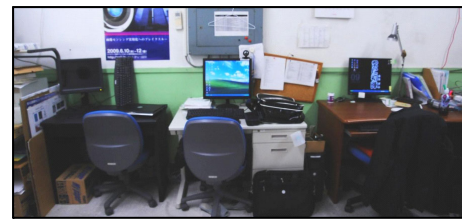
Fig. 15. Detail View of the Passageway Model



Fig. 16. Environment Model of the Room

REFERENCES

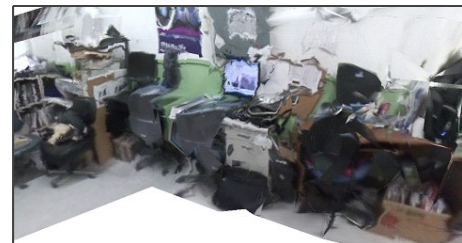
- [1] A. J. Davison: "Real-Time Simultaneous Localisation and Mapping with a Single Camera", Proceedings of the 9th IEEE International Conference on Computer Vision, Vol. 2, pp. 1403-1410, 2003.
- [2] H. Ishiguro, M. Yamamoto and S. Tsuji: "Omni-Directional Stereo", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 257-262, 1992.
- [3] T. Nishimoto, J. Yamaguchi: "Three dimensional measurement using fisheye stereo vision", Proceedings of the Society of Instrument and Control Engineers Annual Conference 2007, 3A05-1, pp. 2008-2012, 2007.
- [4] R. Bunschoten and B. Krose: "Robust Scene Reconstruction from an Omnidirectional Vision System", IEEE Transactions on Robotics and Automation, Vol. 19, No. 2, pp. 351-357, 2003.
- [5] C. Geyer and K. Daniilidis: "Omnidirectional Video", The Visual Computer, Vol. 19, No. 6, pp. 405-416, 2003.
- [6] J. Gluckman and S. K. Nayar: "Ego-motion and Omnidirectional Cameras", Proceedings of the 6th International Conference on Computer Vision, pp. 999-1005, 1998.
- [7] J. Takiguchi, M. Yoshida, A. Takeya, J. Eino and T. Hashizume: "High Precision Range Estimation from an Omnidirectional Stereo System", Proceedings of the 2002 IEEE/RJS International Conference on Intelligent Robots and Systems, pp. 263-268, 2002.
- [8] M. Tomono: "3-D Localization and Mapping Using a Single Camera Based on Structure-from-Motion with Automatic Baseline Selection", Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 3353-3358, 2005.
- [9] J. Meguro, T. Hashizume, J. Takiguchi and R. Kurosaki: "Development of an Autonomous Mobile Surveillance System Using a Network-based RTK-GPS", Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 3107-3112, 2005.
- [10] J. Meguro, Y. Amano, T. Hashizume and J. Takiguchi: "Omni-Directional Motion Stereo Vision Based on Accurate GPS/INS Navigation System", Proceedings of the 2nd Workshop on Integration of Vision and Inertial Sensors, 2005.



(a) Actual Image



(b) Front View



(c) Right Side View



(d) Left Side View

Fig. 17. Other View of the Room Model

- [11] S. Wei, Y. Yagi and M. Yachida: "Building Local Floor Map by Use of Ultrasonic and Omni-Directional Vision Sensor", Proceedings of the 1998 IEEE International Conference on Robotics and Automation, pp. 2548-2553, 1998.
- [12] R. Kawanishi, A. Yamashita, T. Kaneko: "Construction of 3D Environment Model from an Omni-Directional Image Sequence", Proceedings of the 3rd Asia International Symposium on Mechatronics, TP1-3(2), pp. 1-6, 2008.
- [13] J. Y. Bouquet: "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm", OpenCV, Intel Corporation, 2000.
- [14] T. Harada, A. Yamashita and T. Kaneko: "Environment Observation by Structure from Motion with an Omni-directional Camera", Proceedings of International Workshop on Advanced Image Technology 2006, pp. 169-174, 2006.
- [15] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon: "Bundle Adjustment - A Modern Synthesis", Vision Algorithms: Theory & Practice, Springer-Verlag LNCS 1883, 2000.
- [16] A. Nakatsuji, Y. Sugaya, and K. Kanatani: "Optimizing a Triangular Mesh for Shape Reconstruction from Images", IEICE Transactions on Information and Systems, Vol. E88-D, No. 10, pp. 2269-2276, 2005.
- [17] D. Scaramuzza, R. Siegwart: "Appearance-Guided Monocular Omni-directional Visual Odometry for Outdoor Ground Vehicles Robotics", IEEE Transactions on October 2008, Vol. 24, Iss. 5, pp. 1015-1026, 2008.
- [18] A.C. Murillo, J.J. Guerrero, C. Sagüés: "SURF features for efficient robot localization with omnidirectional images", Proceedings of IEEE International Conference on Robotics and Automation 2007, pp. 3901-3907, 2007.