

Three-dimensional Environment Model Construction from an Omnidirectional Image Sequence

Ryosuke Kawanishi, Atsushi Yamashita and Toru Kaneko

Department of Mechanical Engineering, Shizuoka University, Shizuoka, Japan
(Phone : +81-53-478-1604; Email: {f5945016, tayamas, tmtkane}@ipc.shizuoka.ac.jp)

Abstract: When mobile robots execute autonomous tasks, map information is important in path planning and self-localization. In unknown environments, mobile robots must generate their own environmental maps. This paper proposes three-dimensional (3D) environment modeling by a mobile robot. The model is generated from results of 3D measurement and texture information. To measure environmental objects efficiently, the robot uses an image sequence acquired by an omnidirectional camera with wide field of view. The measurement method is based on structure from motion. Triangular meshes are constructed from 3D measurement data. The 3D model is constructed by texture mapping to the triangular mesh, proven by experimental result to be effective.

Keywords: Omnidirectional image sequence, Structure from motion, 3D environment model

1. INTRODUCTION

When mobile robots execute autonomous tasks, map information is important in path planning and self-localization. In unknown environment, mobile robots must construct their own environmental maps.

Three-dimensional (3D) measurement using image data enables map information to be generated [1]. However, images acquired by conventional cameras have a limited field of view [2]. With fisheye [3] and omnidirectional [4] [5] cameras, however, omnidirectional shots providing full environmental measurement and recognition are possible, as shown by Gluckman and Nayar using an omnidirectional camera [6].

In stereo vision using two omnidirectional cameras [7], measurement accuracy depends on the baseline length – the longer the baseline, the better the accuracy. Motion stereo vision uses stereo image pairs taken at different observation points by a single camera, enabling the baseline to be made longer without restriction of robot size [8]. This means motion stereo vision measures more accurately than binocular stereo vision.

Measurement is made even more accurate using laser sensor fusion [9] or geographical positioning systems (GPS) [10] with an omnidirectional camera. However, problems arise in limitation of measurement objects and situations, and it makes system calibration process complex.

Structure from motion (SFM), a type of motion stereo vision, calculates camera movement using corresponding positioning relationships among points in images taken at different observation points, then measures objects in these images.

Our SFM-based proposal extracts and tracks features to get corresponding points in an omnidirectional image sequence. Using positioning relationships, it calculates camera movement and measures environmental objects, generating triangular meshes from measurement data and constructing a 3D environment model by texture mapping to the triangular mesh.

Camera movement must be determined precisely to improve SFM measurement accuracy. Triangular meshes must also be optimized to match the physical environment.

2. OVERVIEW

A mobile robot using an omnidirectional camera such as shown in Fig.1 executes 3D measurement and modeling, acquiring an omnidirectional image sequence as it moves.

As shown in Fig.2, feature points are extracted and tracked to get corresponding points in an omnidirectional image sequence. Using linear estimation, camera movement is calculated using the positioning of corresponding points in two images taken from different observation points and triangulating 3D objects point coordinates. Nonlinear estimation is more precise than linear estimation. Individual measurement data is integrated and a triangular mesh generated from measurement data by Delaunay triangulation, eventually constructs a 3D environment model by texture mapping to the triangular mesh.

3. ALGORITHM

3.1 Acquiring Corresponding Points

To get point correspondence between images in the omnidirectional image sequence, we use the Lucas Kanade tracker algorithm with image pyramid represen-

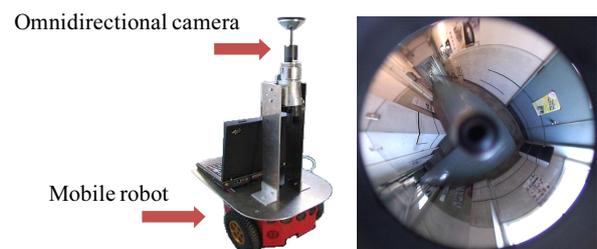


Fig. 1 Mobile Robot and Omnidirectional Image

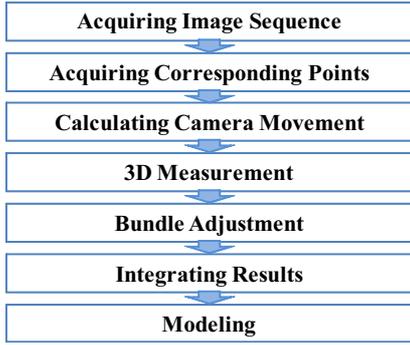


Fig. 2 Proposed Flow

tation [11], as shown in Fig.3, which extracts feature points in the initial image, then tracks them along the sequence. Tracked feature points are regarded corresponding between two images taken at different observation points.

3.2 Calculating Camera Movement

A. Calculating Essential Matrix

We define a unit vector originating from the center of projection to an object point in 3D space as ray vector $\mathbf{r} = [x, y, z]^T$. T is vector or matrix transposition. The omnidirectional camera has a hyperboloid mirror in front of the lens of a conventional camera. As shown in Fig. 4, ray vector \mathbf{r} is directed from the hyperboloid mirror focus to the ray reflection point on the mirror.

Ray vector \mathbf{r} is calculated as follows:

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

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

$[u, v]^T$ are the image coordinates of the feature point. a , b , and c are hyperboloid parameters and f is the image distance between the center of the lens and the image plane.

Matrix \mathbf{E} , the essential matrix, satisfies the following:

$$\mathbf{r}'_i{}^T \mathbf{E} \mathbf{r}_i = 0 \quad (3)$$

where ray vectors $\mathbf{r}'_i = [x_i, y_i, z_i]^T$, $\mathbf{r}_i = [x'_i, y'_i, z'_i]^T$ are those of the corresponding point in two images. The essential matrix contains information about relative positioning and orientation differences between two observation points. Camera rotation matrix \mathbf{R} and translation

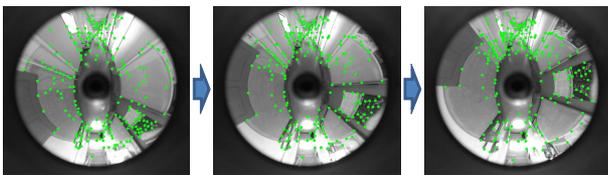


Fig. 3 Feature Extraction and Tracking

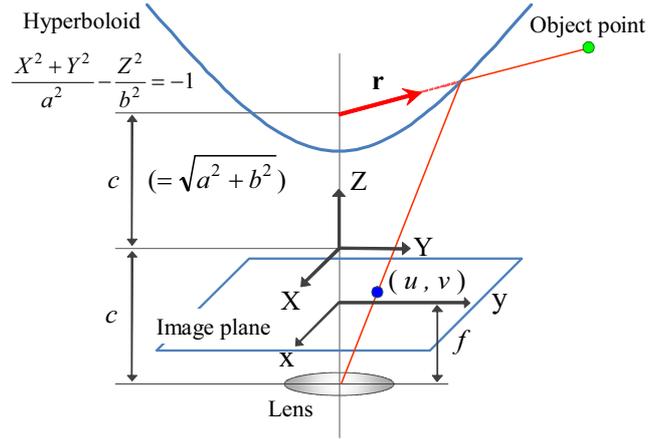


Fig. 4 Ray Vector Calculation

vector \mathbf{t} are calculated from essential matrix \mathbf{E} by singular value decomposition. Calculating essential matrix \mathbf{E} is equivalent to estimating camera movement. 3D object point coordinates are measured using estimated camera movement. With measurement precision depending on camera movement estimation precision, it is vital important to estimate essential matrix \mathbf{E} precisely.

Eq.(3) is transformed as follows:

$$\mathbf{u}^T \mathbf{e} = 0 \quad (4)$$

where

$$\mathbf{u} = [x_i x'_i, y_i y'_i, z_i z'_i, x_i y'_i, y_i y'_i, z_i y'_i, x_i z'_i, y_i z'_i, z_i z'_i]^T$$

$$\mathbf{e} = [e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33}]^T$$

e_{jk} is the row j and column k element of matrix \mathbf{E} .

Essential matrix \mathbf{E} is obtained by solving simultaneous equations for more than eight pairs of corresponding ray vectors, as follows:

$$\min_e \|\mathbf{Ue}\|^2 \quad (5)$$

where $\mathbf{U} = [u_1, u_2, \dots, u_n]^T$. \mathbf{e} is the eigenvector of the smallest eigenvalue of $\mathbf{U}^T \mathbf{U}$, yielding essential matrix \mathbf{E} .

Essential matrix \mathbf{E} is estimated using corresponding points (Section 3.1). However, if these include outliers, estimated camera movement precision is compromised, so, we reject these outliers as detailed below, estimating essential matrix \mathbf{E} using the remaining corresponding points.

B. Outlier Rejection

Not all feature points tracked in the image sequence correspond satisfactorily due to image noise, etc. Mistracked feature points should be rejected, which we do using the random sample consensus (RANSAC) algorithm [12].

Eight feature points – the minimum number for determining essential matrix \mathbf{E} – are selected randomly. Let \mathbf{E}_{rand} be the essential matrix determined using these feature points, and k be the number of feature points satisfying following, where q is a given threshold:

$$|\mathbf{r}'_i{}^T \mathbf{E}_{rand} \mathbf{r}_i| < q \quad (6)$$

Determining essential matrix \mathbf{E}_{rand} and number k are repeated for a predetermined number of times, then we choose the case with the maximum number of k , remove feature points as outliers not satisfying Eq.(6), and calculate essential matrix \mathbf{E} again using remaining feature points.

C. Decision of Feature Point Number

The eight-point algorithm calculates rotation matrix \mathbf{R} and translation vector \mathbf{t} if we get at least eight pairs of corresponding points. Many more than eight pairs are desirable to improve camera movement precisely due to errors in images.

If we extract too many feature points, however, camera movement estimation precision decreases due to the limited number of feature points that can be tracked precisely in an image sequence. Figure 5 shows an example of the relationship between the number of all extracted feature points and the number of outliers. Here, O_k is the number of the outliers when the number of extracted feature points is k .

When most feature points can be correctly tracked, a few outliers included in the feature points are located on the edges of objects. If feature points are located on the edge, correspondence error often occurs, but these feature points are few compared to all extracted feature points cause the Lucas Kanade algorithm evaluates feature points on the corner well, so these feature points are easily rejected using RANSAC if they influence camera movement estimation unfavorably.

If, we extract too many feature points, most added feature points newly have poor features, so we should decide how many feature points are to be used in measurement.

To automatically decide the optimum number of feature points to use in measurement, we define this as the number of feature points including the maximum number of outliers in a range in which camera movement is still estimated precisely enough. From the above, we give Eq.(7) and (8) as follows:

$$\zeta(k) = \frac{w \sum_{i=k}^{k+w} i \cdot O_i - \sum_{i=k}^{k+w} i \sum_{i=k}^{k+w} O_i}{w \sum_{i=k}^{k+w} i^2 - \left(\sum_{i=k}^{k+w} i \right)^2} \quad (7)$$

$$\zeta(k) < t \quad (8)$$

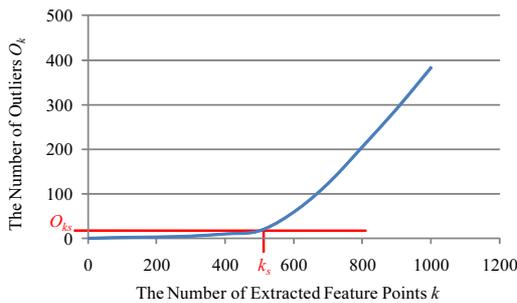


Fig. 5 Relationship Between k and O_k

where O_i is the number of outliers when the number of extracted feature points is i . Equation (7) represents the slope of the straight line calculated by the least squares method, so $\zeta(k)$ shows the increase in the number of outliers when the number of the extracted feature points changes from k to $(k + w)$. $\zeta(k)$ is calculated stably by setting w appropriately. t in Eq.(8) is a given threshold.

The relationship between number of extracted feature points k is not proportional to the increase in number of outliers $\zeta(k)$, as shown in Fig.6. Extracting more than a suitable number of feature points increases the number of outlier drastically while also increasing $\zeta(k)$. In Fig.6, if threshold t in Eq.(8) is t_s , maximum number k satisfying Eq.(8) is k_s . In Fig.5, when the number of extracted feature points is k_s , the number of extracted feature points is large enough to make a detail environment model, and number of outliers O_{k_s} is sufficiently smaller than k_s , i.e., the number of all extracted feature points, so, we calculate maximum k satisfying Eq.(8) as the optimum number of feature points in measurement.

D. Calculation of Camera Movement Parameters

Essential matrix \mathbf{E} is represented by rotation matrix \mathbf{R} and translation vector $\mathbf{t} = [t_x, t_y, t_z]^T$.

$$\mathbf{E} = \mathbf{R}\mathbf{T} \quad (9)$$

\mathbf{T} is a matrix given as follows:

$$\mathbf{T} = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

We calculate \mathbf{R} and \mathbf{T} from the essential matrix \mathbf{E} by singular value decomposition.

3.3 3D Measurement

3D coordinates of object points are calculated by triangulation with two cameras in the geometrical relationship given by rotation matrix \mathbf{R}_m and translation vector \mathbf{t}_m , where m is the number of observation points. We calculate 3D coordinates of object point $\mathbf{p}_{m,i}$ (i -th feature point) using rotation matrix \mathbf{R}_m , translation vector \mathbf{t}_m , and two ray vectors $\mathbf{r}_{1,m,i}$ and $\mathbf{r}_{2,m,i}$ as shown in Fig.7.

Measurement accuracy is lower when an object point is close to the baseline or far from the camera, so, we evaluate measurement accuracy based on the differentiation of measurement result $\mathbf{p}_{m,i}$, defining an error vector

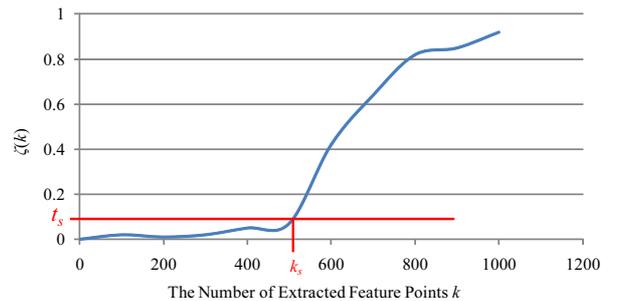


Fig. 6 Relation between k and $\zeta(k)$

calculated by Eq.(10) as the evaluation of measurement accuracy.

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

$[u_{m,i}, v_{m,i}]^T$ and $[u'_{m,i}, v'_{m,i}]^T$ are i -th image coordinates of the two feature points at the m -th observation point. We select measurement results satisfying Eq.(11), where h is a threshold:

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

3.4 Bundle Adjustment

As explained above, we estimate camera movement based on measurement errors in each feature point, using bundle adjustment to give a nonlinear least squares solution by minimizing the sum of reprojection errors [13]. To calculation reprojection errors, we use the result of camera movement (Section 3.2) as initial parameters. Reprojection error is the difference between original feature point and reprojected coordinates. If reprojection error is low, camera movement is estimated is highly precise. Reprojected feature point coordinates are calculated as follows:

$$\begin{bmatrix} u \\ v \\ -c + f \end{bmatrix} = d' \begin{bmatrix} dx \\ dy \\ dz + 2c \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -c \end{bmatrix} \quad (12)$$

$$d = \frac{cz' + b\sqrt{x'^2 + y'^2 + z'^2}}{\left(\frac{b}{a}\right)^2 (x'^2 + y'^2) - z'^2} \quad (13)$$

$$d' = \frac{f}{dz' + 2c} \quad (14)$$

We define the sum of feature reprojection errors as follows:

$$E_m = \sum_i r_{m,i} |\mathbf{x}_{m,i} - \mathbf{x}'_{m,i}|^2 \quad (15)$$

E_m is the sum of feature reprojection errors at the m -th observation point, $r_{m,i}$ the weight factor for the i -th feature point calculated evaluating error in each feature point, $\mathbf{x}_{m,i}$ the original feature point coordinate, and $\mathbf{x}'_{m,i}$ the reprojected coordinate. The norm of vector $\mathbf{g}_{m,i}$ calculated by Eq.(10) evaluates following error:

$$r_{m,i} = \frac{1}{\|\mathbf{g}_{m,i}\|} \quad (16)$$

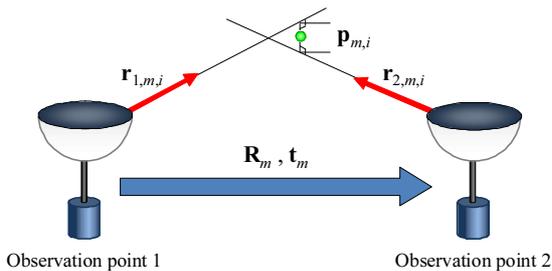


Fig. 7 Calculation of 3D Coordinates

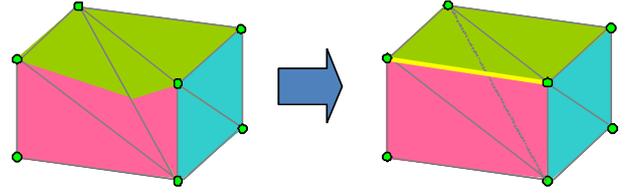


Fig. 8 Triangular Mesh Optimization

$\mathbf{g}_{m,i}$ is a vector expressing the measurement accuracy of the i -th feature point at the m -th observation point. We add a larger weight to feature points with smaller vectors, i.e., small error.

3.5 Integrating Results

To integrate the above measurement results, we solve scale ambiguity among individual measurements by scale matching [14]. After scale matching, more than two measurement results may show the same object point, so we integrate these into one object point voting for voxels divided in 3D space. The 3D coordinate of the object point is the coordinate of the voxel having the largest value.

3.6 Modeling

Delaunay triangulation often generates a triangular mesh contradicting physical shape by failing to consider measurement object shape. We apply triangular optimization [15] to the triangular mesh (Fig.8). The method adapts the triangular mesh to the physical shape by detecting texture distortion. Texture mapping to the triangular mesh yields a 3D environment model.

4. EXPERIMENTS

In experiments, we decide thresholds q , h , w , and t by trial and error. Using an omnidirectional camera on a manipulator, we treat manipulator movement as true camera movement by having manipulator form a square, then estimated camera movement 4 times between each apex of the square. Camera start and end locations are the same. The square is 350 millimeters on the side, as shown in Fig.9. Threshold q is set to 0.03, h to 0.15, w to 50, and t to 0.4.

Table 1 shows estimated camera locations. Figures 10 and 11 graph Table 1, where camera locations A, B, C, D, and E correspond to apexes of the square traced by

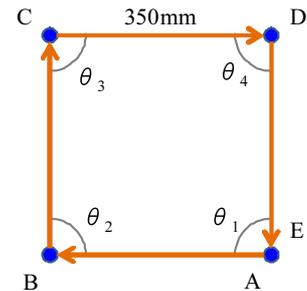


Fig. 9 Camera Movement – Translation Alone

Table 1 Camera Location Result – Translation Alone

	Without Our Proposal			with Our Proposal		
	X	Y	Z	X	Y	Z
A	0	0	0	0	0	0
B	-0.995	0.071	-0.026	-0.994	0.087	-0.032
C	-1.032	0.040	0.958	-1.018	0.034	0.979
D	0.080	-0.654	1.015	-0.039	-0.040	1.007
E	-0.077	-0.036	0.029	-0.031	0.031	0.029
A-E Distance	0.089			0.052		

Table 2 Distance Results

	Without Our Proposal	With Our Proposal
AB	1.000	1.000
BC	0.986	1.015
CD	1.315	0.983
DE	1.176	0.982
Average	1.119	0.995
Standard Deviation	0.135	0.0134
Maximum Error from Average	0.195	0.020

Table 3 Direction Result

	Without Our Proposal	With Our Proposal
θ_1 [deg]	89.5	90.8
θ_2 [deg]	88.4	89.5
θ_3 [deg]	115.4	91.4
θ_4 [deg]	78.9	88.3
Average [deg]	93.0	90.0
Standard Deviation [deg]	13.9	1.22
Maximum Error [deg]	15.4	0.6

the manipulator in Fig.9. Results not using our proposal show unnatural camera movement not tracing a square, whereas results using our proposal nearly trace a square. In Table 2, our proposal makes standard deviation and maximum error one tenth that of results not using our proposal. In Table 3, results using our proposal make standard deviation and maximum error 25 times smaller than results not using our proposal.

For camera movement including rotation and translation, Fig.12 shows true camera movement in this experiment. The blue axis is camera rotation axis θ , the green axis camera rotation axis ϕ , and the red axis camera rotation axis ψ . A manipulator with an omnidirectional camera traced arcs AB, BC and CD with rotation. Camera location and positioning at end point D are equivalent to those at start point A. Camera positioning change is \mathbf{R} (θ [deg], ϕ [deg], ψ [deg]) rotation around each rotation axis θ , ϕ , and ψ , and camera positioning change \mathbf{t} (X [mm], Y [mm], Z [mm]), which is the translation parallel to each axis X , Y , and Z . Movement along arcs AB, BC and CD are \mathbf{t} (0, 350, -350) and \mathbf{R} (-90, 0, 0), \mathbf{t} (350, -350, 0) and \mathbf{R} (0, 90, 0), and \mathbf{t} (0, -350, 350) and \mathbf{R} (0, -90, 90). Threshold q is set to 0.02, h to 0.08, w to 50, and t to 0.4.

Table 4 shows estimated camera locations at A, B, C, and D, and the distance from A to D, where the distance between A and B is 1. Figures 13 and 14 graph Table 4, where blue marks A, B, C, and D correspond to camera

locations A, B, C, and D in Fig.12. The distance between A and D using our proposal is one tenth that not using our proposal. Table 5 shows estimated results of camera rotation from A to B, B to C, and C to D. Standard deviation and maximum error are calculated using true camera movements. Result using our Proposal has smaller standard deviation and maximum error than that not using our proposal. Estimation errors are due to outliers and low-accuracy measurement points, which showing that our proposal rejects outliers and selects highly accurate measurement points, demonstrating its effectiveness in measurement including data with errors.

We measured two environments shown in Fig.15(a) a passageway including an L-shape corner and shown in Fig.15 (b) a room. We acquired environment image sequences (10 fps) using a mobile robot that moved at 10 cm/s and used an omnidirectional camera. Sequence image were 1920×1080 pixels. In the passageway experiment, threshold q is set to 0.02, h to 0.2, w to 50, and t to 0.8. In the room experiment, threshold q is set to 0.02, h to 0.8, w to 50, and t to 0.4.

Figure 16 shows integrated passageway measurement results. Blue marks in the results show the robot trajectory and red marks measurement data. Camera movement estimated using our proposal (Section 3.2-B and C, Eq.(10) and (11) in Section 3.3, and Section 3.4) has higher precision than that not using it because outliers and low-accuracy measurement points are almost all removed in Fig.16 (a), indicating our proposal measured the passageway shape correctly as shown by the lack of little contradiction between camera movement and integrated measurement data.

We constructed a 3D environment model of the passageway using measurement results using our proposal, as shown in Fig.16 (b). Figure 17 shows a bird's-eye view and Fig.18 actual environmental images at left and details at right. Although the model contains some tex-

Table 4 Camera Location Results – Rotation and Translation

	Without Our Proposal			With Our Proposal		
	X	Y	Z	X	Y	Z
A	0	0	0	0	0	0
B	-0.102	0.794	-0.633	-0.086	0.778	-0.621
C	0.779	0.062	-0.559	0.643	0.135	-0.610
D	-0.305	-0.568	0.324	-0.008	-0.069	-0.030
A-D Distance	0.721			0.076		

Table 5 Camera Positioning Change Results

	Without Our Proposal			With Our Proposal		
	θ [deg]	ϕ [deg]	ψ [deg]	θ [deg]	ϕ [deg]	ψ [deg]
A-B	-86.9	-8.5	-3.3	-86.0	-7.5	-3.6
B-C	-2.6	80.1	8.3	-0.9	81.8	5.5
C-D	1.2	-82.4	98.0	-0.8	-85.2	89.4
Standard Deviation	6.5			4.6		
Maximum Error	9.9			8.2		

ture distortions, results show that our proposal constructs a model equivalent to the physical shape of the measured object.

Figure 19 shows integrated measurement results. Camera trajectories in (a) and (b) are mostly the same. We get many good feature points tracked correctly in the experimental environment – the room in Fig.15 – because many room objects had much texture. Even so, they are not always suitable for measurement. In Fig.19 (a), many measurement points have low-accuracy, but, in Fig.19 (b), our proposal (Eq.(10) and (11) in Section 3.3) rejected most low-precision measurement results.

We constructed a 3D environment model of the room by using measurement results from our proposal (Fig.19 (b)). Figure 20 shows actual environmental images at left and views of the 3D environment model at right. The object's shape is recognizable in an arbitrary view. Although this model is rough due to sparse measurement, results show that our proposal measures the room shape.

5. CONCLUSIONS

We have proposed calculating camera movement precisely by selecting good feature points through removing most outliers and low-accuracy measurement points, thereby improving camera movement calculation accuracy for constructing maps. Applying triangular optimization adapts triangular meshes to physical shape by detecting texture distortion. The effectiveness of our proposal was demonstrated in experimental results.

To improve measurement accuracy, we plan to calculate camera movement by considering more than three observation points. For the environment model, we must use texture mapping that considers omnidirectional image distortion. We should also calculate and recognize shape edges, planes, and curved surfaces of objects in environments based on the relationship between measurement data and texture distortion possessed by constructed models.

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/RSJ 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.
- [11] J. Y. Bouguet: "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm", OpenCV, Intel Corporation, 2000.
- [12] M. A. Fischler and R. C. Bolles: "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", Communications of the ACM, Vol. 24, No. 6, pp. 381-395, 1981.
- [13] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon: "Bundle Adjustment -A Modern Synthesis", Vision Algorithms: Theory & Practice, Springer-Verlag LNCS 1883, 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] 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.

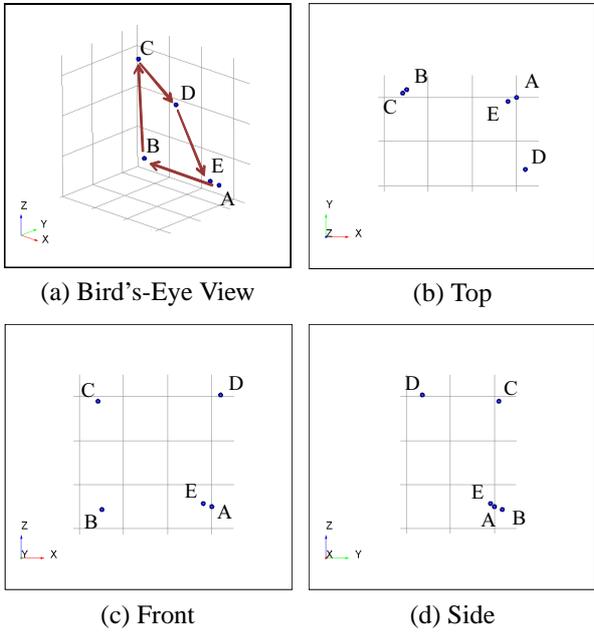


Fig. 10 Results of Camera Movement Not Using Our Proposal – Translation Alone

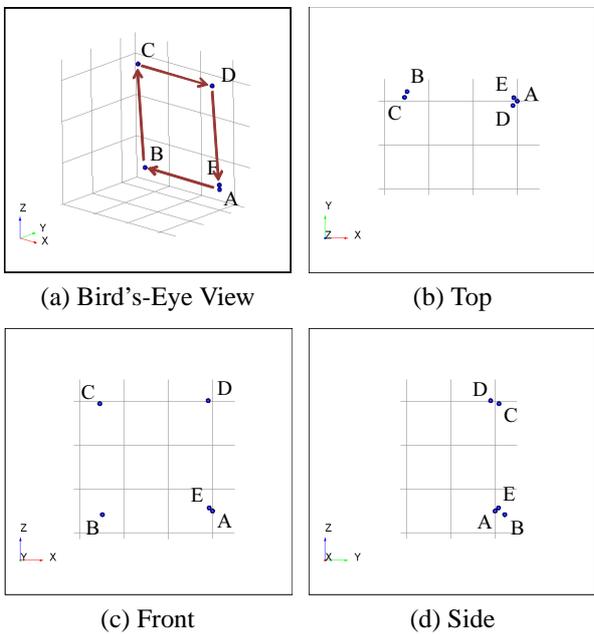


Fig. 11 Results of Camera Movement Using Our Proposal – Translation Alone

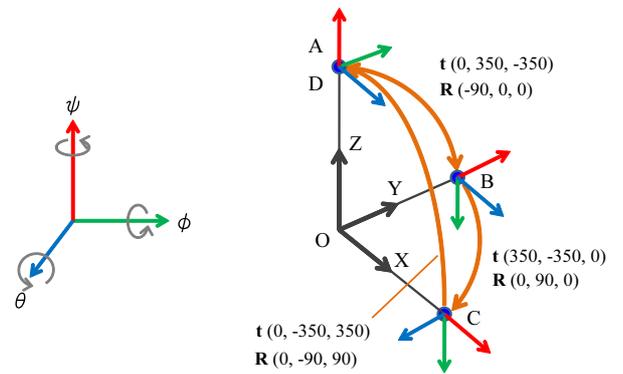


Fig. 12 Camera Movement with Rotation and Translation

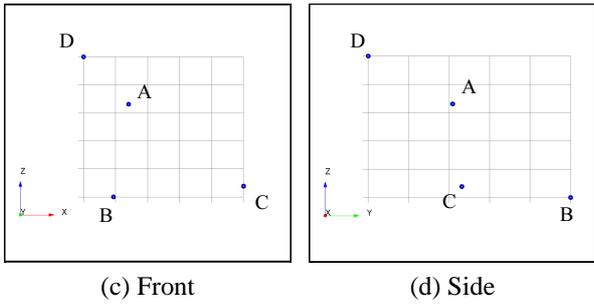
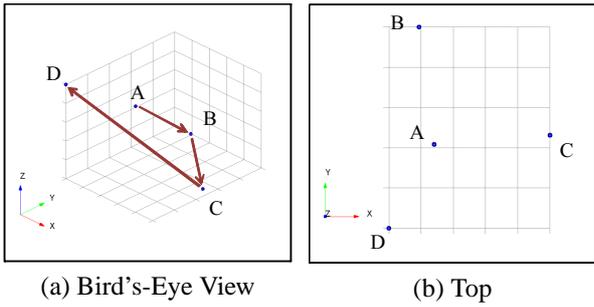


Fig. 13 Results of Camera Movement Not Using Our Proposal – Rotation and Translation

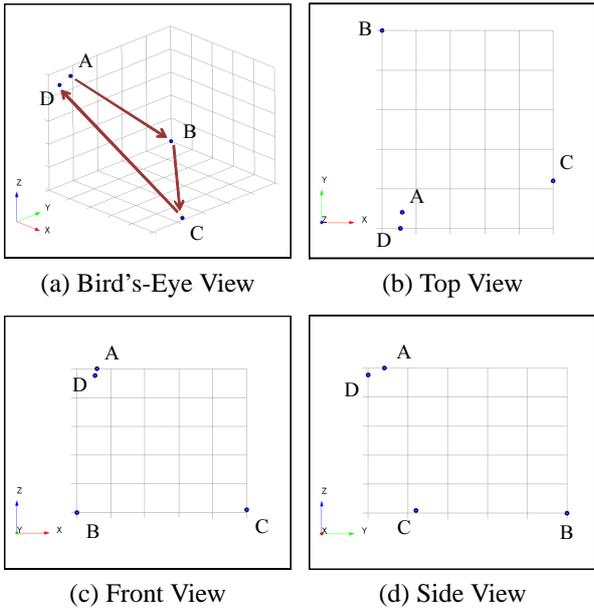


Fig. 14 Result of Camera Motion Estimation Using Our Proposal – Rotation and Translation

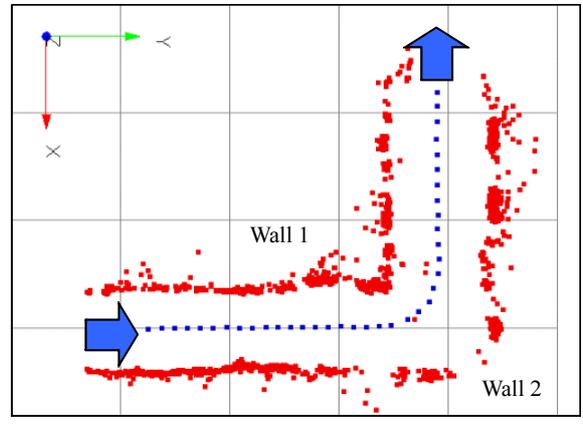
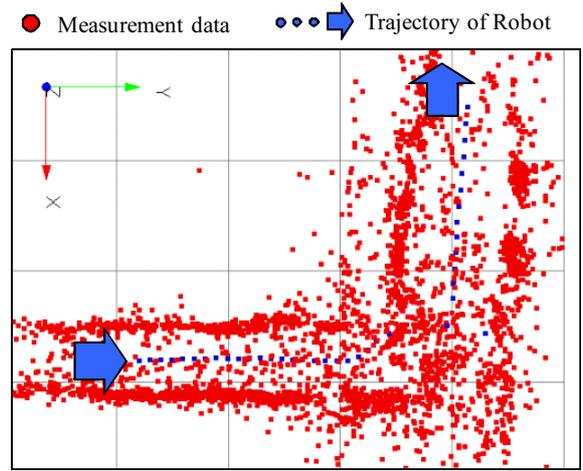
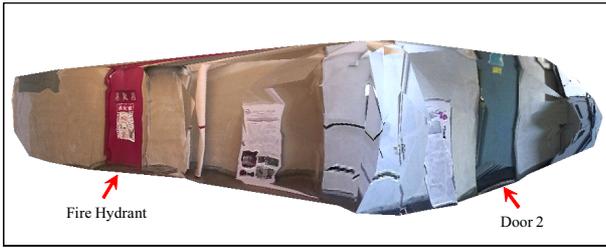


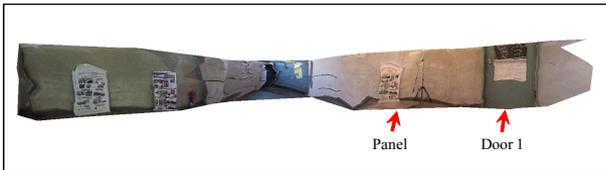
Fig. 16 Integrated Measurement Data of Passageway



Fig. 15 Experiment Environments



(a) Wall 1



(b) Wall 2

Fig. 17 Bird's-eye View of 3D Environment Model



(a) Door 1



(b) Panel

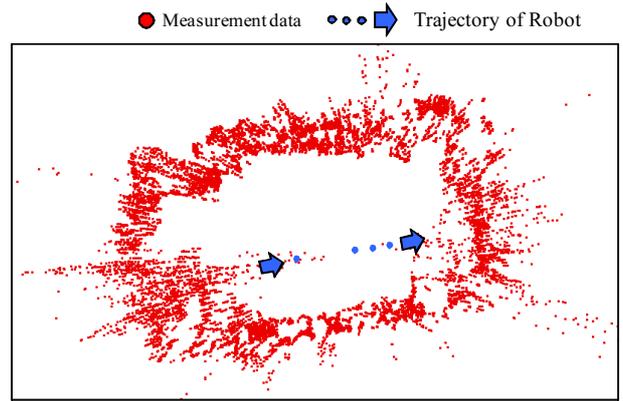


(c) Fire Hydrant

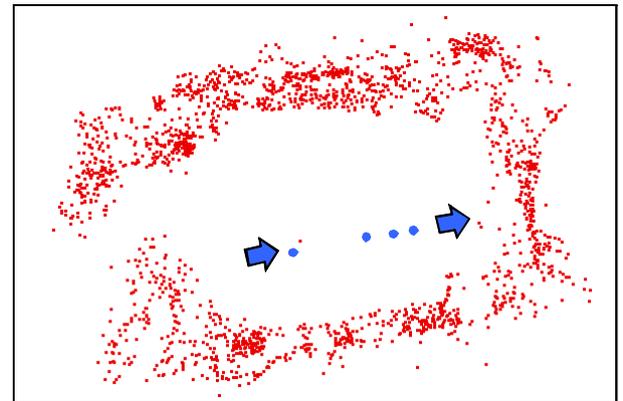


(d) Door 2

Fig. 18 Detail View. Left: Actual Image, Right: Constructed Model



(a) Result Not Using Our Proposal



(b) Result Using Our Proposal

Fig. 19 Inregrated Measurement Data of Room



(a)



(b)



(c)

Fig. 20 3D Room Model. Left: Actual Image, Right: Constructed Model