

Line-based Camera Movement Estimation by Using Parallel Lines in Omnidirectional Video

Ryosuke kawanishi, Atsushi Yamashita, Toru Kaneko and Hajime Asama

Abstract—In this paper, we propose an efficient estimation method of an omnidirectional camera movement. The proposed method is based on Structure from Motion utilizing a constraint of parallel lines. In an environment having man-made structures, parallel lines can be extracted from an omnidirectional image easily and constantly, because of its wide field of view. Parallel lines provides a valuable constraint for camera movement estimation. The proposed method can estimate 3-D camera movements by solving one degree of freedom problem three times without regard to the number of viewpoints. Experimental results show the effectiveness of our proposed method.

I. INTRODUCTION

To estimate camera movement accurately and efficiently is an essential assignment for scene reconstruction by monocular stereo, such as an approach based on Structure from Motion (SfM) [1], [2] or vSLAM [3], [4]. Camera movement estimation from an image sequence acquired by a single camera is difficult, because a camera movement matrix, such as the essential matrix, has at least 6 degrees of freedom (DOFs) (rotation is 3 DOFs and translation is 3 DOFs) and is a non-linear matrix. The processing cost will be high and the calculation of the optimal camera matrices will be complex if viewpoints to be estimated increase.

A limited field of view of cameras equipped with a typical lens makes it difficult to estimate camera movement, too. Self-localization method by monocular stereo often needs feature tracking (KLT tracker [5], SIFT [6], and so on) to obtain correspondence between different viewpoints. However, features will be lost easily due to camera swing. Consequently, the camera movement will be limited if we use a such camera for scene reconstruction.

Therefore, we use an omnidirectional camera equipped with a hyperboloid mirror. An omnidirectional camera is suitable for self-localization [7], because the acquired image has a panoramic field of view (Fig. 1). Self-localization and scene reconstruction method by using an omnidirectional camera have been proposed [8], [9], [10].

In previous monocular stereo, there are some approaches by using feature points [2], [3], [8], [9], [10], straight lines



Fig. 1. The omnidirectional camera which we use is shown in the left figure. The camera attaches a hyperboloid mirror on the front of the camera lens. An omnidirectional image is shown in the right figure. The image has a 360-degree horizontal field of view.

[11], [12], [13] or both features [1]. Point-based methods have the benefit of fast calculation by the linear solution (8-point algorithm [14], 5-point algorithm [2], and so on). However, there may be not sufficient number of feature points in indoor environments. On the other hand, the method needs to solve higher DOF problem than a point-based method, because a line-based method has to use more than 3 viewpoints for camera movement estimation. There are linear computation algorithms for a line-based SfM by using trifocal tensor [15], [16]. However, these methods are only for image triplets.

We propose an efficient camera movements estimation method by using correspondences of parallel lines between not less than three images. The constraint obtained from parallel lines is useful for reduction in the degree of freedom of the camera movement estimation. As a previous method, SLAM by using parallel lines and their vanishing point [17], SfM in urban environment (building scene) [18], rotation estimation by video compass [19] and so on have been proposed. However, the computation time of these method increases exponentially as the number of features and viewpoints increase [17], [18]. In addition, these methods assume that a camera movement is only horizontal movement [18], [19].

The proposed method can estimate 3-D camera movements of more than 3 viewpoints efficiently. Rotation matrices and translation vectors estimation are divided in two phases. Each phase is solved as a 1 DOF problem without regard to the number of viewpoints and features. Consequently, although these are non-linear problems, the calculation cost is extremely low. The proposed method can find the global optimal solution easily even if an initial value near to the ground truth is not given. The effectiveness of our proposed method is shown in experimental results.

The proposed method needs at least 3 images and 6 lines. Three of these lines have to be parallel. Parallel lines are easily extracted from a man-made structure in an indoor

This work was not supported by any organization

R. Kawanishi and Toru Kaneko are with Department of Mechanical Engineering, Faculty of Engineering, Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu-shi, Shizuoka 432-8561, Japan f5945016@ipc.shizuoka.ac.jp, tmtkane@ipc.shizuoka.ac.jp

A. Yamashita and H. Asama is with Department of Precision Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan yamashita@robot.t.u-tokyo.ac.jp, asama@robot.t.u-tokyo.ac.jp

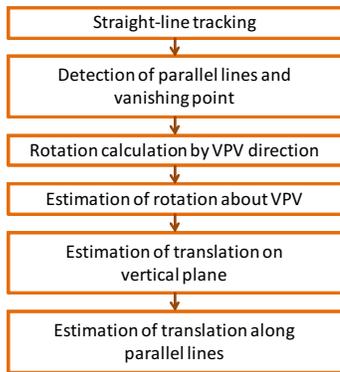


Fig. 2. Procedure of our proposed method. Estimation of rotation matrices and translation vectors are divided in two phases. Each phase is solved as a 1 DOF problem without regard to the number of viewpoints and features.

environment, because an omnidirectional camera has a wide field of view. The others must have different direction from the parallel lines. A line which connects two feature points can be used as the other line. Therefore, the assumption is relevant for general indoor environments. An omnidirectional camera is calibrated in advance. The procedure of our proposed method is shown in Fig. 2.

Straight-lines are extracted and tracked along an omnidirectional video. Parallel lines and the vanishing point are detected from these lines. The vector directed to the vanishing point from the viewpoint is calculated. It is called a VPV in this paper. Camera rotation estimation is divided in two phases. In the first phase, the method calculates a camera rotation matrix which makes a VPV at each viewpoint have at the same 3-D direction. In the second phase, a rotation matrix about a VPV is estimated. If a rotation matrix between two viewpoints is determined, rotation matrices at the other viewpoints are calculated by solving a quartic function about the rotation angle. Therefore, in the proposed method, rotation estimation can be solved as a 1 DOF problem without regard to the number of viewpoints and features.

Camera translations are estimated by two phases. In the first phase, translations in a plane are estimated. The plane is vertical to 3-D direction of parallel lines. If a translation direction between two viewpoints is determined, translations at the other viewpoints are calculated by solving a simultaneous equation. Therefore, the estimation is a 1 DOF problem about the translation direction. In the second phase, translations along 3-D direction of parallel lines are estimated. In this phase, if a translation between two viewpoints is determined, translations at the other viewpoints are obtained by algebraic calculations. Consequently, the proposed method estimates camera movements by solving 1 DOF problems.

II. COORDINATE SYSTEM OF OMNIDIRECTIONAL CAMERA

The coordinate system of our omnidirectional camera is shown in Fig. 3. A hyperboloid mirror reflects a ray heading to image coordinates (u, v) from the camera lens. In this paper, the reflected ray is called a ray vector. The extension lines of all ray vectors intersect at the focus of

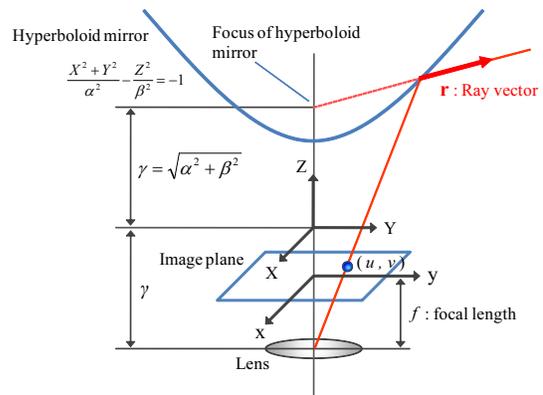


Fig. 3. The coordinate system of the omnidirectional camera. Ray vector is defined as a unit vector which starts from the focus of a hyperboloid mirror.

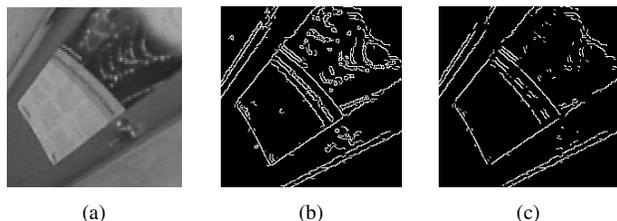


Fig. 4. Edge segment extraction. (a) Input image. (b) Detected canny edge points. (c) Edge points which are line-like.

the hyperboloid mirror. The ray vector is calculated by the following equations.

$$\mathbf{r} = \begin{bmatrix} \lambda(u - c_x)p_x \\ \lambda(v - c_y)p_y \\ \lambda f - 2\gamma \end{bmatrix}, \quad (1)$$

$$\lambda = \frac{\alpha^2 \left(f\sqrt{\alpha^2 + \beta^2} + \beta\sqrt{u^2 + v^2 + f^2} \right)}{\alpha^2 f^2 - \beta^2 (u^2 + v^2)}, \quad (2)$$

where c_x and c_y are the center coordinates of the omnidirectional image, p_x and p_y are pixel size, f is the focal length of a camera lens, α , β and γ are hyperboloid parameters.

III. STRAIGHT-LINE TRACKING

Straight-lines are extracted from a distorted omnidirectional image. The proposed method obtains edge points detected by Canny edge detector [20]. An example of edge point detection is shown in Fig. 4(a) and (b).

To separate each straight-lines, corner points are rejected as shown in Fig. 4(c). Corner points are detected by using two eigenvalues of the Hessian of the image. If the ratio of eigenvalues is high enough, the edge point is regarded as line-like. Other edge points are rejected as corner points. The ratio is set to 10 by the trial-and-error method.

A least square plane is calculated from ray vectors of segmented edge points. If the edge segment consists of a straight-line, these ray vectors are located on a plane (Fig. 5). Therefore, an edge segment which has a small least square error is regarded as a straight-line. The proposed method can extract straight-lines, even if an edge segment looks like

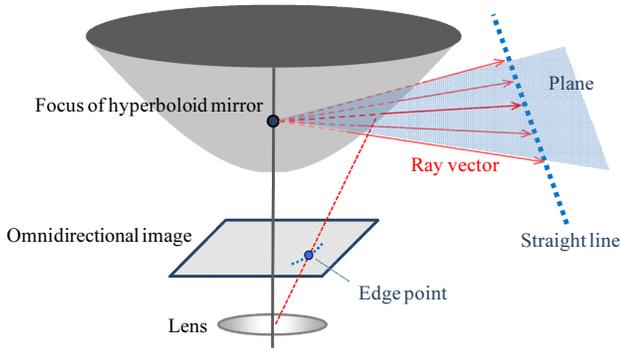


Fig. 5. The relationship between a straight-line and a ray vector. Ray vectors which belongs to the line are located on the plane including the line.

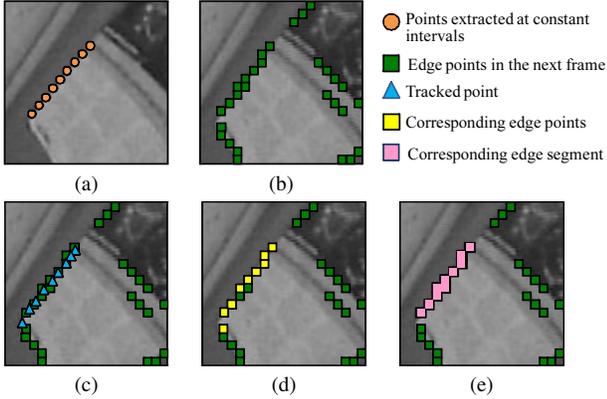


Fig. 6. Searching for a corresponding edge segment in the next frame. (a) Points extracted at constant intervals. (b) Edge segments in the next frame. (c) Points (a) are tracked between the current frame and the next frame. (d) Corresponding edge points. (e) Corresponding edge segment.

a curve in an omnidirectional image. If over half the edge points of the edge segment i satisfy the following equation, the segment is determined as a straight-line.

$$(\mathbf{r}_{i,j}^T \mathbf{n}_i)^2 < l_{th}, \quad (3)$$

where l_{th} is a threshold. $\mathbf{r}_{i,j}$ is a ray vector heading to the edge point j included in the line i . \mathbf{n}_i is the normal vector of the least square plane calculated from the line i . \mathbf{n}_i is a unit vector. The vector is called NV in this paper. In the detection, edge points which do not constitute the line are rejected as noise by RANSAC [21]. The threshold l_{th} is determined from the image resolution.

Straight-lines are tracked along the omnidirectional image sequence. The proposed method obtains points located on a straight-line. Points located on the line are extracted at constant intervals (Fig. 6(a)). Edge segments are extracted in the next frame (Fig. 6(b)). The points extracted in Fig. 6(a) are tracked to the next frame by KLT tracker (Fig. 6(c)). The edge point closest to the tracked point is selected as a corresponding edge point (Fig. 6(d)). The edge segment which has the maximum number of corresponding edge points is regarded as a corresponding edge segment (Fig. 6(e)). If an edge segment corresponds to several lines, a line which has

larger number of corresponding edge points is selected.

Matching point search on a line has the aperture problem [22]. However, it is not difficult for the proposed method to obtain corresponding edges, because it does not require point-to-point matching. By continuing the above processes, straight-lines are tracked along the omnidirectional image sequence.

IV. DETECTION OF PARALLEL LINES AND VANISHING POINT

Parallel lines and their vanishing point are detected from tracked lines. A VPV \mathbf{v}_c and NV \mathbf{n}_c about parallel lines at a viewpoint c satisfy the following equation.

$$\mathbf{n}_c^T \mathbf{v}_c = 0. \quad (4)$$

Three lines are required for parallel lines detection from an image. The proposed method selects three lines from tracked lines randomly. A VPV is calculated from selected lines by solving (5) by the least squares method.

$$\sum_i^{n_l} (\mathbf{n}_{i,c}^T \mathbf{v}_c)^2 \rightarrow \min, \quad (5)$$

where n_l is the number of lines. If selected lines satisfy the following equation at all of an input image sequence, these are regarded as parallel lines.

$$\sum_i^3 \mathbf{n}_{i,c}^T \mathbf{v}_c < p_{th}, \quad (6)$$

where p_{th} is a threshold. Lines which are regarded as parallel each other are integrated. A line group which has the maximum number of lines is used for the following process as parallel lines. The VPV \mathbf{v}_c at the viewpoint c is calculated from the integrated parallel lines by (5).

V. ROTATION ESTIMATION

Camera rotation estimation is divided in two phases. In the first phase, the method calculates a camera rotation matrix which makes a VPV at each viewpoint have the same 3-D direction. In the second phase, a rotation matrix about a VPV is estimated by using at least 3 lines. These lines must have different 3-D direction from parallel lines.

A. Rotation calculation by VPV direction

This phase requires VPVs only. A VPV at each viewpoint should have the same 3-D direction in the world coordinate, because a vanishing point is theoretically at an infinite distance away from viewpoints. Therefore, the proposed method calculates a rotation matrix \mathbf{R}_c^m satisfying the following equation.

$$\mathbf{v}_{c_0} = \mathbf{R}_c^m \mathbf{v}_c. \quad (7)$$

where \mathbf{R}_c^m is a rotation matrix between the initial viewpoint c_0 and a viewpoint c . In this paper, the initial camera coordinate system is equal to the world coordinate system. \mathbf{R}_c^m is calculated as a rotation matrix about a vector \mathbf{m}_c by Rodrigues rotation formula. The rotation angle θ_c is

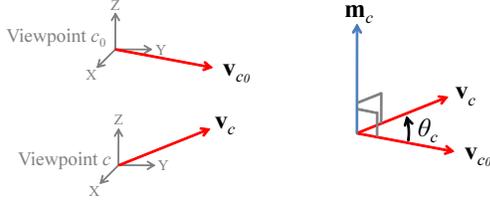


Fig. 7. The relationship among VPVs \mathbf{v}_{c_0} , \mathbf{v}_c , a rotation axis vector \mathbf{m}_c and the rotation angle θ_c .

calculated A relationship between \mathbf{m}_c and θ_c are shown in Fig. 7. They are defined as the following equations.

$$\mathbf{m}_c = \mathbf{v}_{c_0} \times \mathbf{v}_c, \quad (8)$$

$$\theta_c = \arccos(\mathbf{v}_{c_0} \cdot \mathbf{v}_c). \quad (9)$$

The 3-D direction of parallel lines and the VPV are the same. Therefore, the VPV \mathbf{v}_{c_0} also means 3-D direction of parallel lines in the following explanation.

B. Estimation of rotation around parallel line axis

In the second phase, a rotation matrix about the VPV \mathbf{v}_{c_0} is estimated. This phase needs at least three lines. The 3-D direction of these lines must not be equal to the VPV \mathbf{v}_{c_0} .

In the proposed method, unknown parameter of 3-D rotation remains only a rotation \mathbf{R}_c^v about the VPV \mathbf{v}_{c_0} , because the other two parameters can be obtained from a constraint of VPV direction. Therefore, this phase estimates a rotation matrix \mathbf{R}_c^v , namely, the rotation angle ϕ_c (Fig. 8).

The true rotation matrix \mathbf{R}_c between the initial viewpoint c_0 and a viewpoint c is defined as the following equation.

$$\mathbf{R}_c = \mathbf{R}_c^m \mathbf{R}_c^v. \quad (10)$$

The true rotation matrix \mathbf{R}_c and 3-D line direction \mathbf{d}_i satisfy the following equation.

$$(\mathbf{R}_c^T \mathbf{n}_{i,c})^T \mathbf{d}_i = 0, \quad (11)$$

where $\mathbf{n}_{c,i}$ is the NV of the line i at the viewpoint c . \mathbf{d}_i is a unit vector. If a rotation angle ϕ_c is given, the 3-D direction \mathbf{d}_i of the line i is calculated by the following equation.

$$\mathbf{d}_i = \mathbf{n}_{c_0} \times (\mathbf{R}_c^T \mathbf{n}_{i,c}). \quad (12)$$

By using the 3-D line direction, a rotation matrix \mathbf{R}_k between the initial viewpoint c_0 and the other viewpoint k is calculated by solving the following equation.

$$e_{rot}(\phi_k) = \sum_i^{n_l} \left| (\mathbf{R}_k^T \mathbf{n}_{i,k})^T \mathbf{d}_i \right|^2 \rightarrow \min. \quad (13)$$

where, n_l is the number of non-parallel lines. Although the function is non-linear, it can be solved easily because $e_{rot}(\phi_k)$ is just a quartic function about ϕ_k . Consequently, if a rotation angle ϕ_c at a viewpoint c is given, rotation angles ϕ_k at the other viewpoints k are determined. The proposed

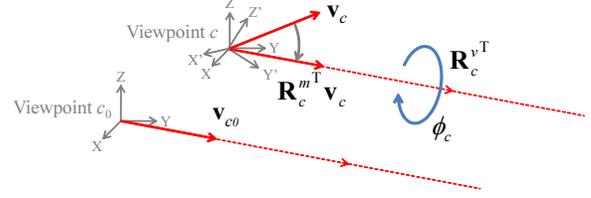


Fig. 8. Rotation about VPV \mathbf{v}_{c_0} .

optimization of rotations is represented as the following equation.

$$E_{rot}(\phi_c) = \sum_k^{n_c} e_{rot}(\phi_k) \rightarrow \min, \quad (14)$$

where n_c is the number of viewpoints. In the proposed method, rotation estimation is 1 DOF problem about the rotation angle ϕ_c without regard to the number of viewpoints and features.

In the following explanation, \mathbf{v} expresses the VPV \mathbf{v}_{c_0} . \mathbf{v} also means 3-D direction of parallel lines.

VI. TRANSLATION ESTIMATION

Camera translations are estimated by two phases. In the first phase, translations on the plane vertical to \mathbf{v} are estimated. In the second phase, translations directed along parallel lines are estimated. As same as the rotation estimation, translations can be optimized by solving 1 DOF problems.

A. Estimation of translation on vertical plane

In the first phase, translations on the plane vertical to the VPV \mathbf{v} are estimated. This phase requires parallel lines.

Translations in the plane and locations of parallel lines are optimized simultaneously. Here, the method introduces basis vectors \mathbf{a} and \mathbf{b} ($\mathbf{a} \perp \mathbf{b}$, $\mathbf{a} \perp \mathbf{v}$, and $\mathbf{b} \perp \mathbf{v}$). A unit vector $\mathbf{g}_{i,c}^p$ directed to the parallel line i from the viewpoint c are calculated by (15). The vector $\mathbf{g}_{i,c}$ is perpendicular to the VPV.

$$\mathbf{g}_{i,c}^p = \mathbf{v} \times (\mathbf{R}_c^T \mathbf{n}_{i,c}). \quad (15)$$

By using these vectors, \mathbf{a} and \mathbf{b} elements of the true translations can be estimated. A translation vector \mathbf{t}_c^p between the initial viewpoint c_0 and a viewpoint c , the location \mathbf{l}_i^p of parallel lines i and a vector $\mathbf{g}_{i,c}^p$ satisfy the following equation. The relationship among these vectors is shown in Fig. 9.

$$\delta_{i,c} \mathbf{g}_{i,c}^p + \mathbf{t}_c^p - \mathbf{l}_i^p = 0, \quad (16)$$

where $\delta_{i,c}$ is a fixed number which means the depth of the line at the viewpoint. $\delta_{i,c}$ is calculated by the following equation.

$$\delta_{i,c} = \frac{(\mathbf{t}_c^p - \mathbf{l}_i^p)^T \mathbf{g}_{i,c}^p}{\mathbf{g}_{i,c}^p{}^T \mathbf{g}_{i,c}^p}. \quad (17)$$

Here, the translation \mathbf{t}_c^p is expressed by (18).

$$\mathbf{t}_c^p = \mathbf{a} \cos \psi_c + \mathbf{b} \sin \psi_c, \quad (18)$$

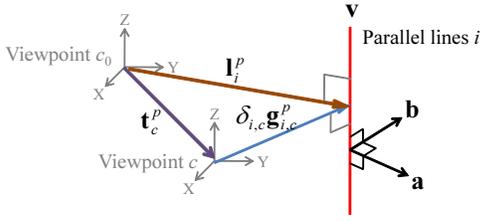


Fig. 9. The relationship among VPV \mathbf{v} , basis vector \mathbf{a} and \mathbf{b} , a line location \mathbf{l}_i^p , a vector $\mathbf{g}_{i,c}^p$ and a translation \mathbf{t}_c^p on the plane.

where ψ_c means translation direction from the initial viewpoint c_0 to the viewpoint c . The absolute scale is unknown in the SfM approach. Thus, the distance between these two viewpoints is set to 1 in the proposed method. When the direction ψ_c is given, the locations of parallel lines are calculated by the following equation.

$$\mathbf{l}_i^p = (\zeta_{i,c_0} \mathbf{g}_{i,c_0}^p + \eta_{i,c} \mathbf{g}_{i,c}^p + \mathbf{t}_c^p) / 2, \quad (19)$$

where ζ_{i,c_0} and $\eta_{i,c}$ are fixed factors which mean the depth of the line from each viewpoint. Fixed factors ζ_{i,c_0} and $\eta_{i,c}$ are calculated as factors satisfying the following expression.

$$\|\zeta_{i,c_0} \mathbf{g}_{i,c_0}^p - \eta_{i,c} \mathbf{g}_{i,c}^p - \mathbf{t}_c^p\|^2 \rightarrow \min. \quad (20)$$

By using the locations of parallel lines, translations at the other viewpoint k are calculated by minimizing the following equations.

$$E_{t_1}(\lambda_k, \mu_k) = \sum_i^{n_l} \left\| \delta_{i,k} \mathbf{g}_{i,k}^p + \mathbf{t}_k^p - \mathbf{l}_i^p \right\|^2, \quad (21)$$

$$\mathbf{t}_k^p = \lambda_k \mathbf{a} + \mu_k \mathbf{b}. \quad (22)$$

When the line location \mathbf{l}_i^p , namely, the translation direction ψ_c is given, $E_{t_1}(\lambda_k, \mu_k)$ can be solved as simultaneous equations about λ_k and μ_k . Therefore, translations on the plane vertical to parallel lines are optimized by minimizing the following function about ψ_c .

$$E_{t_1}(\psi_c) = \sum_k^{n_c} E_{t_1}(\lambda_k, \mu_k). \quad (23)$$

B. Estimation of translation along parallel lines

In the second phase, translations directed along parallel lines are optimized by the method based on Bundle adjustment [23]. At least 3 lines are required in this phase. The 3-D direction of the lines used for the estimation must be different from VPV. The true translation vector \mathbf{t}_c is represented as the follows:

$$\mathbf{t}_c = \mathbf{t}_c^p + \omega_c \mathbf{v}, \quad (24)$$

where ω_c is an unknown parameter about translation distance to be estimated in this phase. Translations and line locations

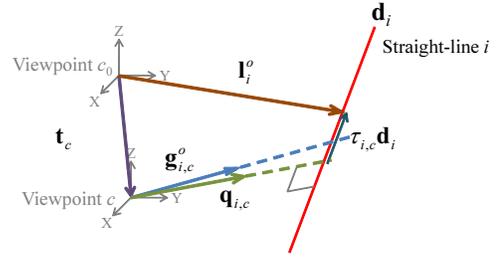


Fig. 10. Reprojection error of straight-line.

are optimized by minimizing the sum of reprojection errors as shown in Fig. 10 and (25)-(27).

$$E_{t_2} = \sum_i^{n_l} (1 - \mathbf{q}_{i,c}^T \mathbf{g}_{i,c}^o)^2, \quad (25)$$

$$\mathbf{q}_{i,c} = \frac{\mathbf{l}_i^o - \mathbf{t}_c + \tau_{i,c} \mathbf{d}_i}{\|\mathbf{l}_i^o - \mathbf{t}_c + \tau_{i,c} \mathbf{d}_i\|}, \quad (26)$$

$$\tau_{i,c} = \frac{(\mathbf{t}_c - \mathbf{l}_i^o)^T \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{d}_i}, \quad (27)$$

where $\mathbf{q}_{i,c}$ is a unit vector crossed at a right angle to the straight-line i . $\mathbf{g}_{i,c}^o$ is a unit vector heading to the line from the viewpoint c . The vector is calculated by the same way as (15). If there are no errors, these vectors will be the same. However, in fact, these have different direction because of various errors. The angle error is almost the same to a reprojection error.

When ω_c , namely, the translation from the initial viewpoint c_0 to the viewpoint c is determined, 3-D line locations \mathbf{l}_i^o are calculated in the same way as (16). By using the locations, (25) at the other viewpoint k can be solved as a quartic function about ω_k . Therefore, the translation estimation is a non-linear 1 DOF problem about ω_c . The optimum translations are estimated by minimizing the following equation.

$$E_{t_2}(\omega_c) = \sum_k^{n_c} E_{t_2}(\omega_k). \quad (28)$$

VII. EXPERIMENT

We demonstrate proposed camera movement estimation by using simulation data. In these experiments, the true values of NVs $\mathbf{n}_{i,c}$ acquired at each viewpoint c are given. It is known that which lines are parallel. The vector heading to their vanishing point is calculated from given normal vectors. The camera movement includes 3-D rotations and translations. In these experiments, we estimate camera movements by exhaustive searches without using any preconditions.

At first, we verified that the proposed method can estimate camera movements from 6 lines. Three lines of these are parallel. The others have different direction. The number of viewpoints is 10. The position relationship between the viewpoints and lines is shown in Fig. 11. Red, yellow and orange axes show a camera coordinate system at each viewpoint. Parallel lines are represented by green color. Other

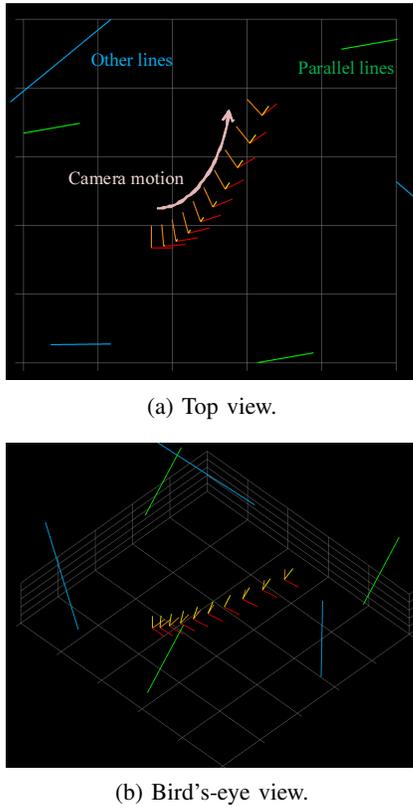


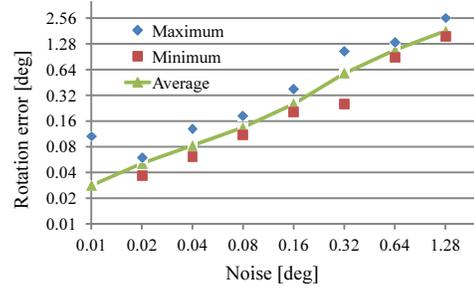
Fig. 11. Estimated camera movement and line locations.

lines are represented by blue color. The estimation errors of camera movement and line measurement are within a round-off error of the computation in this experiment.

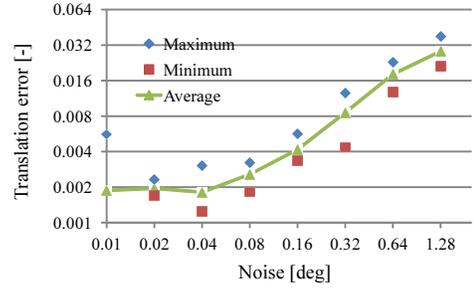
We verified the robustness of the proposed method in noisy data. In this experiment, noisy NVs are given. The noise means an angle error between a given vector and the true one. The noise follows normal gaussian distribution. We evaluated estimation errors of camera rotations and translations by using noisy data including 0.01 to 1.28 degrees angle errors on average. Input data is 40 lines including 20 parallel lines acquired at 20 viewpoints. A different noise is added to a NV randomly in 10 times trial runs.

The estimation results are shown in Fig. 12. The rotation error in Fig. 12 (a) means an angle errors between the axis of estimated camera coordinate system and the true one. The translation error in Fig. 12 (b) is a distance between an estimated camera location and the true one. These values are the average of 10 times trial runs. In this experiment, translation distance between the beginning viewpoint and the end viewpoint is set to 1, because translation scale is indefinite in the SfM approach. The translation error mean a distance error between the true location and the estimated location.

These results show that the proposed method performs well in noisy data, because these estimation errors are nearly within the given noise. The proposed method can find the global optimal solution easily because the proposed method can estimate camera movement by solving 1 DOF problem



(a) Rotation.



(b) Translation.

Fig. 12. Estimation errors with noisy data.

three times. It contributes improvement of robustness of self-localization.

We verified the proposed method by using real images. 200 images are acquired by a mobile robot equipped with an omnidirectional camera. The movement distance is about 1.5 meters. The image size is 800×600 . An input image is shown in Fig. 13. The experimental environment is an indoor (texture-less hallway). This experiment is done with off-line processing. CPU is Intel Core i7 975 (3.33 GHz).

Extracted lines are shown in Fig. 14. Blue lines show parallel lines. Red lines show the other lines. Correspondence of 10 parallel lines and 5 other lines were obtained from the input omnidirectional video. The computation time of line tracking is 25 ms per frame. That of parallel lines detection is 35 ms.

The estimation result of camera movement and line measurement is shown in Fig. 15. This figure shows a top view of hallway. Although camera rotations and translations at 200 viewpoints are estimated, this figure shows 20 positions of all viewpoints for easier viewing. An average of the reprojection errors is within 6 pixels. The major cause of the error is correspondence accuracy of lines. The computation time of rotation estimation is 2 ms. That of translation estimation is 16 ms.

VIII. CONCLUSION

In this paper, we proposed an efficient estimation method of 3-D camera movement. Camera rotations and translations are estimated as a reasonable 1 DOF problem. Although these are non-linear problems, the calculation cost is extremely low and to find the global optimal solution is easy

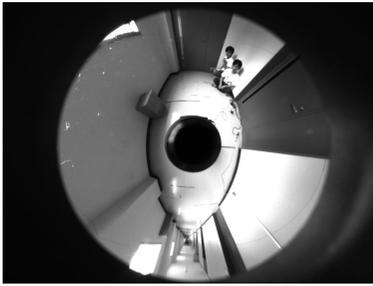


Fig. 13. Input image. The image sequence is acquired in an indoor environment (texture-less hall way).

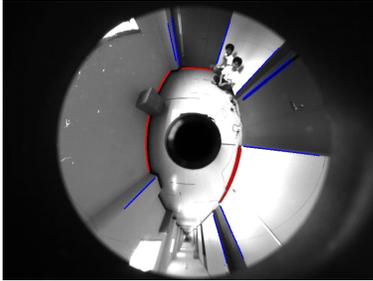


Fig. 14. Extracted lines. Blue lines show parallel lines. Red lines show the other lines.

even if an initial value near to the ground truth is not given. Experiments show the effectiveness of the proposed method.

As future works, efficient and robust search for obtaining the solution should be considered. We should verify the proposed method in a large environment.

IX. ACKNOWLEDGMENTS

This work was in part supported by MEXT KAKENHI, Grant-in-Aid for Young Scientist (A), 22680017.

REFERENCES

[1] D. Nister: "Reconstruction From Uncalibrated Sequences with a Hierarchy of Trifocal Tensors", *Proceedings of the 6th European Conference on Computer Vision*, Vol. 1, pp. 649-663, 2000.

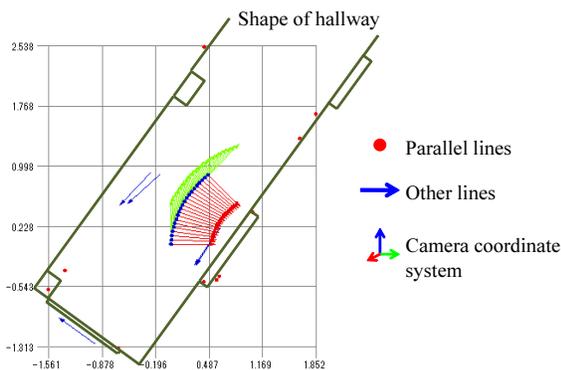


Fig. 15. Estimation result of camera movement and line measurement.

[2] D. Nister: "An Efficient Solution to the Five-Point Relative Pose Problem", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 6, pp. 756-770, 2004.

[3] 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.

[4] M. Tomono: "3D Object Mapping by Integrating Stereo SLAM and Object Segmentation Using Edge Points", *Advances in Visual Computing, Lecture Notes in Computer Science*, Vol. 5875, pp. 690-699, 2009.

[5] J. Shi and C. Tomasi: "Good Features to Track", *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

[6] D. G. Lowe: "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[7] J. Gluckman and S. K. Nayar: "Ego-motion and Omni-directional Cameras", *Proceedings of the 6th International Conference on Computer Vision*, pp. 999-1005, 1998.

[8] 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.

[9] C. Geyer and K. Daniilidis: "Omnidirectional Video", *The Visual Computer*, Vol. 19, No. 6, pp. 405-416, 2003.

[10] A. Murillo, J. Guerrero, and C. Sagues: "SURF Features for Efficient Robot Localization with Omnidirectional Images", *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 3901-3907, 2007.

[11] A. Bartoli and P. Sturm: "Multi-View Structure and Motion from Line Correspondences", *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 207-212, 2003.

[12] A. Bartoli and P. Sturm: "Structure-from-motion using lines: Representation, Triangulation, and Bundle Adjustment", *Computer Vision and Image Understanding*, Vol. 100, Issue 3, pp. 416-441, 2005.

[13] P. Smith, I. Reid and A. Davison: "Real Time Monocular SLAM with Straight lines", *Proceedings of the 2006 British Machine Vision Conference*, pp. 17-26, 2006.

[14] R. Hartley: "In Defense of the Eight-point Algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 6, pp. 580-593, 1997.

[15] P. H. S. Torr and A. Zisserman: "Robust Parameterization and Computation of the Trifocal Tensor", *Image and Vision Computing*, Vol. 15, Issue 8, pp. 591-605, 1997.

[16] R. Hartley: "Lines and Points in Three Views and the Trifocal Tensor", *International Journal of Computer Vision*, Vol. 22, No. 2, pp. 125-140, 1997.

[17] M. Bosse, R. Rikoski, J. Leonard, S. Teller: "Vanishing Points and 3D Lines from Omnidirectional Video", *Proceedings of the 2002 International Conference on Image Processing*, Vol. 3, pp. 513-516, 2002.

[18] G. Schindler, P. Krishnamurthy and F. Dellaert: "Line-Based Structure from Motion for Urban Environments", *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 846-853, 2006.

[19] G.L. Mariottini and D. Prattichizzo: "Uncalibrated Video Compass for Mobile Robots from Paracatadioptric Line Images", *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 226-231, 2007.

[20] J. F. Canny: "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 679-698, 1986.

[21] M. A. Fischler and R. C. Bolles: "Random Sample Con-sensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Carto-graphy", *Communications of the ACM*, Vol. 24, No. 6, pp. 381-395, 1981.

[22] K. Nakayama and G. Silverman: "The Aperture Problem – I. Spatial Integration of Velocity Information Along Contours", *Vision Research*, Vol. 28, No. 6, pp. 747-753, 1988.

[23] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon: "Bundle Adjustment – A Modern Synthesis", *Vision Algorithms: Theory and Practice, Lecture Notes in Computer Science*, Vol. 1883 pp. 298-372, 2000.