Floor-wall boundary estimation by ellipse fitting

Tomonari MASUZAKI[1], Yasuyuki SUGAYA[1], and Kenichi KANATANI[2]
[1]Toyohashi University of Technology, Aichi Japan, {masuzaki, sugaya}@iim.cs.tut.ac.jp
[2]Okayama University, Okayama Japan, kanatani2013@yahoo.co.jp

***Abstract*—In order to realize autonomous robot navigation, we propose a method for detecting the boundary between a floor and a wall by extracting an ellipse projected from a robot-mounted projector. We note that a circular cone of light projected onto the region where the floor and the wall meet produces two intersecting ellipses in the image. We detect the two ellipses from the image and estimate the line that separates them, which we regard as the floor-wall boundary. We also propose a new method for fitting an ellipse to an edge point sequence which may contain outliers, i.e., non-elliptic arcs. Using real images, we show that our proposed method can detect the floor-wall boundary even when the detected edge point sequence contains outliers that have complicated forms.**

***Index Terms*—Autonomous robot navigation, Ellipse fitting, Outlier detection**

I. Introduction

In order to realize autonomous robot navigation, many methods have been proposed for detecting walls and obstacles. They are classified into three approaches: the use of range sensors [16], the use of computer vision techniques [1], [12], [14], [15], and their combinations [13].

A well-known computer-vision-based approach is SLAM, which simultaneously estimates the robot position and generates a 3-D map of the environment. It is based on 3-D reconstruction from the images taken by a robot-mounted stereo cameras. Hence, the accuracy of the resulting 3-D reconstruction deteriorates if we try to extract feature points on texture-less objects like walls and to track them,

In this paper, we propose a method for detecting the boundary between the floor and the wall by extracting an ellipse projected from a robot-mounted projector. If a circular cone of light is projected onto the region where the floor and the wall meet, we observe two ellipses in the image. We can regard the line that separates them as the floor-wall boundary. In order to extract the two ellipses, we extract edge points and fit an ellipse to them. For this purpose, it is rather an advantage of our method that the floor and the wall have low-texture. However, we may also extract edge points that do not belong to elliptic arcs. Hence, we need to consider the existence of such outliers.

Many methods have been proposed for extracting elliptic arcs from an image and for fitting an ellipse to the extracted arcs [6], [17], [19]. Ellipse parameters can be obtained even if when only a part of the ellipse is observed. Hence, it is expected that a robust system can be built by projecting light and doing ellipse fitting.

Many methods exist for fitting an ellipse to the obtained elliptic arcs [8], [9], [2], [10], [7], [4], [18], [11]. However, most of them do not consider the presence of non-elliptic arcs, which we call *outliers*, in the input data. RANSAC is a well known framework for dealing with outliers [3]. However, RANSAC requires a lot of iterations before obtaining the final result, because it involves random sampling. Yu et al. [19] detected and removed outlier points by considering the fitting residuals Iteratively applying this, they fitted an ellipse to the remaining inliers. However, the number of inliers decreases in the course of the iterations. This sometimes decreases the accuracy of the resulting fit.

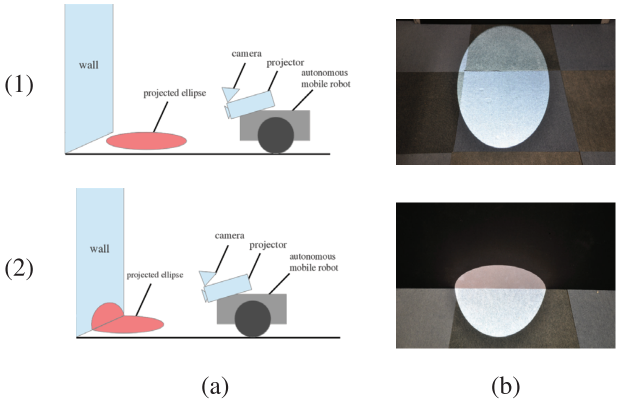

Fig. 1. (a) Proposed autonomous mobile robot. We project a circular cone of light onto the floor and capture the image of it. (b) If the light is projected onto the region where the floor and the wall meet, two intersecting ellipses are produced and the line that separates them can be detected as the floor-wall boundary.

In this paper, we propose a new method for simultaneously fitting an ellipse and detecting outliers. Assuming that input data are a spatially connected sequence of edge points, we segment it into partial arcs by considering the ellipse fitting residuals and detect inlier arcs by computing the curvature of the residual graph. Our fitting method has several advantages over existing methods. Although our method involves iterations, the number of iterations is much less than those of RANSAC and Yu's method.

II. Proposed method

We estimate the floor-wall boundary by using a projector and a camera mounted on a mobile robot. If a circular cone of light is projected onto a flat surface, we observe an ellipse in the image (Fig. 1(1)). If it is projected onto the region where the floor and the wall meet, we observe two intersecting ellipses in the image (Fig. 1(2)). In the latter case, the line that separates them is the floor-wall boundary. Thus, extracting two ellipses from the image, we can detect the floor-wall boundary.

978-1-4673-7338-8/15/$31.00 ©2015 IEEE

Since the projector and the camera are fixed to the robot, the shape of the ellipse we observe does not change if the floor is completely horizontal and its position relative to the robot is the same. Thus, the parameters of the expected ellipse can be computed in advance. Hence, we can detect the floor-wall boundary by extracting the part projected onto the wall and by estimating the line that separates it from the part projected onto the floor. Our floor-wall boundary detecting method is summarized as follows:

Initialization procedure:

1) Project a circular cone of light onto the floor and capture the image.
2) Extract edge points and fit an ellipse to them.

Floor-wall boundary detection procedure:

1) Project a circular cone of light and capture the image.
2) Extract an edge point sequence and segment it into multiple arcs, and choose the arc that has the largest number of points.
3) Remove those edge points that (approximately) belong to the ellipse obtained in the initialization procedure.
4) Fit an ellipse to the remaining edge points by the ellipse fitting method described below.
5) Analytically compute the line that separates the two fitted ellipses.

In the following, we summarize our method for analytically computing the separating line and describe the details of our ellipse fitting techniques.

III. Floor-Wall boundary detection from ellipses

A. Ellipse expression

Curves represented by a quadratic equations in x and y in the form

$$Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Ey) + f_0^2F = 0, \quad (1)$$

are called *conics*, which include ellipses, parabolas, hyperbolas, and their degeneracies such as two lines [5].

In Eq. (1), f_0 is a constant that has the order of the image size for stabilizing finite length numerical computation[1]. If we define 3-D vector $\boldsymbol{x}$ and 3×3 matrix $\boldsymbol{Q}$ in the form

$$\boldsymbol{x} = \begin{pmatrix} x \\ y \\ f_0 \end{pmatrix}, \qquad \boldsymbol{Q} = \begin{pmatrix} A & B & D \\ B & C & E \\ D & E & F \end{pmatrix}. \quad (2)$$

Eq. (1) is written as

$$(\boldsymbol{x}, \boldsymbol{Q}\boldsymbol{x}) = 0, \quad (3)$$

where $(\boldsymbol{a}, \boldsymbol{b})$ denotes the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

B. Intersection lines of two ellipses

Given two ellipses $(\boldsymbol{x}, \boldsymbol{Q}_1\boldsymbol{x}) = 0$ and $(\boldsymbol{x}, \boldsymbol{Q}_2\boldsymbol{x}) = 0$, the equation

$$(\boldsymbol{x}, (\boldsymbol{Q}_1 + \lambda\boldsymbol{Q}_2)\boldsymbol{x}) = 0, \quad (4)$$

describes, for an arbitrary constant λ, a quadratic curve or its degeneracy that passes through all the intersections (if there are any) of the two conics. If we select a λ such that

$$|\boldsymbol{Q}'| = 0, \quad \boldsymbol{Q}' = \boldsymbol{Q}_1 + \lambda\boldsymbol{Q}_2, \quad (5)$$

Eq. (4) degenerates into two real or imaginary lines. If the two ellipses intersect at four points, the two lines that pass through them can be written as

$$A'x + (B' \pm \sqrt{B'^2 - A'C'})y + \left(D' - \frac{B'D' - A'E'}{\sqrt{B'^2 - A'C'}}\right) f_0 = 0, \quad (6)$$

where $A', B', ...$ are the elements of $\boldsymbol{Q}'$ corresponding to Eq. (2). If the two ellipses are tangent to each other at two points, the line that passes through the two tangent points is given by

$$A'x + B'y + D'f_0 = 0. \quad (7)$$

C. Floor-Wall boundary detection

The two ellipses resulting from projection of circular cone of light on to the region where a floor surface meets a wall are tangent to each other at two points; the tangent lines at them corresponds the tangent planes to the circular cone of light. The line that passes through the two tangent points are regarded as the floor-wall boundary. However, the detected ellipses from the image may not be necessarily tangent to each other at two points due to the inaccuracy of the ellipse fitting procedure. In order to cope with this, we compute the constant λ according to the following rules, noting that Eq. (5) should theoretically have a double root if the two ellipses are tangent to each other:

1) If Eq. (5) has three real roots, take the λ to be the midpoint of the two closest roots.
2) If Eq. (5) has one real root and two complex conjugate roots, take λ to be the real part of the complex conjugate pair.

We compute $\boldsymbol{Q}'$ from the resulting λ and estimate the floor-wall boundary as Eq. (7).

IV. Ellipse fitting with outliers

A. Ellipse fitting

Our task is to compute the coefficients $A, ..., F$ so that the ellipse of Eq. (1) passes through the detected points (x_α, y_α), $\alpha = 1, ..., N$, as closely as possible. For a point sequence (x_α, y_α), $\alpha = 1, ..., N$, we define 6-D vectors

$$\begin{aligned} \boldsymbol{\xi}_\alpha &= (x_\alpha^2,\, 2x_\alpha y_\alpha,\, y_\alpha^2,\, 2f_0x_\alpha,\, 2f_0y_\alpha,\, f_0^2)^\top, \\ \boldsymbol{\theta} &= (A,\, B,\, C,\, D,\, E,\, F)^\top. \end{aligned} \quad (8)$$

The condition that (x_α, y_α) satisfies Eq. (1) is written as

$$(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta}) = 0. \quad (9)$$

Since vector $\boldsymbol{\theta}$ has scale indeterminacy, we normalize it to unit norm: $\|\boldsymbol{\theta}\| = 1$.

[1] We set f_0 = 600.

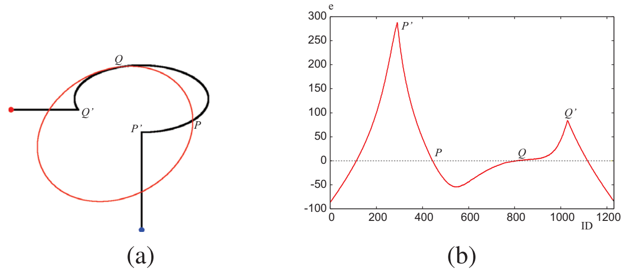

Fig. 2. (a) An ellipse fitted to a point sequence which includes non-elliptic arcs. A black line is an input point sequence. A blue and a red point are a start and an end point of the input point sequence. (b) Fitting residual. The horizontal axis shows the index of the points, which starts from the blue point to the red point shown in (a). The vertical axis shows the signed fitting residual, whose sign is computed by the left-hand side of Eq. (9).

Since Eq. (9) is not exactly satisfied in the presence of noise, we compute a $\boldsymbol{\theta}$ such that $(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta}) \approx 0$, α = 1, ..., N. For computing a $\boldsymbol{\theta}$ that is close to its true value, we need to consider the statistical properties of noise. The standard model is to regard the noise in (x_α, y_α) as an independent Gaussian random variable of mean 0 and standard deviation σ. Then, the covariance matrix of the vector $\boldsymbol{\xi}_\alpha$ has the form $\sigma^2 V_0[\boldsymbol{\xi}_\alpha]$, where

$$V_0[\boldsymbol{\xi}_\alpha] = 4\begin{pmatrix} \bar{x}_\alpha^2 & \bar{x}_\alpha\bar{y}_\alpha & 0 & f_0\bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha\bar{y}_\alpha & \bar{x}_\alpha^2+\bar{y}_\alpha^2 & \bar{x}_\alpha\bar{y}_\alpha & f_0\bar{y}_\alpha & f_0\bar{x}_\alpha & 0 \\ 0 & \bar{x}_\alpha\bar{y}_\alpha & \bar{y}_\alpha^2 & 0 & f_0\bar{y}_\alpha & 0 \\ f_0\bar{x}_\alpha & f_0\bar{y}_\alpha & 0 & f_0^2 & 0 & 0 \\ 0 & f_0\bar{x}_\alpha & f_0\bar{y}_\alpha & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{10}$$

which we call the *normalized covariance matrix* [7]. It depends only on the true position $(\bar{x}_\alpha, \bar{y}_\alpha)$ of the data point (x_α, y_α). In numerical computation, we replace the true values by the data values[2].

B. Overview of the proposed ellipse fitting

We assume that input points are a spatially connected curve and that fitted ellipse intersects the curve at multiple points (Fig. 2(a)). We segment the input curve into partial arcs at these intersection points. We compute the variation of the tangent angle to the graph of the fitting residual, which we simply call *error curvature* in this paper, and judge if each arc is an inlier or an outlier based on this error curvature.

The algorithm of our method is summarized as follows:

1) Fit an ellipse to a point sequence by Fitzgibbon's method [4].
2) Compute the sign of the left-hand side of Eq. (9) for all the points and segment the point sequence into partial arcs at the points across which the computed sign changes.
3) For each segmented arc, detect the point where the residual takes a maximum and compute its error curvature ϕ at this point.
4) Go to Step (a) if it is the first ellipse fitting, else go to Step (b).
    a) Select an inlier arc which has the smallest value ϕ among those arcs whose arc lengths are longer than a threshold[3]. Extend it to adjacent arcs if their end points correspond to the peak of the arc. Then, we fit an ellipse to these selected arcs.
    b) Select the arcs whose error curvature ϕs are smaller than a threshold[4] $\hat{\phi}$ and fit an ellipse to them.
5) Repeat the procedures from Step 2 to Step 4 until the number of inliers does not change.

The principle of our method is nearly equivalent to the curvature-based segmentation and outlier detection. However, our method is more efficient, because our method computes the curvature of the residual graph at only one point where the residual takes a maximum in each segmented arc.

In Fig. 2, the input point sequence is segmented into five partial arcs by the fitted ellipse. The residual value of the arc PQ, which consists only of an elliptic arc, smoothly changes around the peak value. On the other hand, we can see that the residual graph has a peaky shape over the arcs consisting of non-elliptic arcs.

Moreover, we can see that the arcs PP' and QQ', which are connected to the elliptic arc PQ, are also elliptic arcs. The points P' and Q' are the peak points of the partial arcs adjacent to the elliptic arc PQ. Therefore, if we use not only the detected inlier arc but also the adjacent arcs like the arcs PP' and QQ' for ellipse fitting, we can effectively fit a correct ellipse.

As discussed before, if the selected arc is an elliptic arc, the adjacent arcs are also elliptic arcs. Therefore, we can effectively fit a correct ellipse if we use those arcs. However, if we select a non-elliptic arc as an inlier and add adjacent outlier arcs to fit an ellipse, we cannot fit a correct ellipse.

For this reason, in the first iteration of our algorithm, we select among the arcs that are sufficiently long the one whose error curvature ϕ at its peak point is the smallest. We regard it as a reliable inlier and extend it to the adjacent partial arcs. After the first iteration, we select all arcs whose error curvature ϕs are smaller than a threshold $\hat{\phi}$. We do not extend those arcs, because the adjacent arcs can be non-elliptic arcs if the selected inlier arcs approximately belong to the correct ellipse.

In the following sections, we describe the details of our method.

C. Division of the point sequence

The left-hand side of Eq. (9) at the point p_α has a different sign outside and inside the ellipse $\boldsymbol{\theta}$. Using this fact, we segment the input point sequence $\{p_\alpha\}$ into partial arcs at those points p_α which have different signs from their neighboring points.

[2] We have confirmed that this does not affect the results of our experiments.

[3] We set the threshold to be 5% of the number of input edge points.

[4] We set $\hat{\phi} = 80°$.

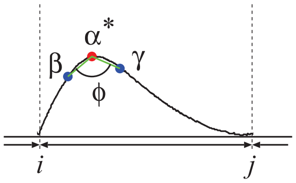

Fig. 3. Error curvature ϕ of the peak point α^* for a partial arc.

Numerically, however, the value $(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})$ may not be exactly zero even if the point p_α lies on the fitted ellipse $\boldsymbol{\theta}$; the points lying on the fitted ellipse may irregularly change their signs. Hence, these points can be divided in very short arcs. To avoid this, we regard those partial arcs whose fitting error is close to zero as elliptic arcs, and judge that they are inlier arcs without computing its curvature. We call such arcs *tangent arcs*.

D. Inlier arc selection

After segmenting the point sequence into multiple arcs, we detect the point where the fitting error takes its maximum in each arc and compute its error curvature at this point. If the computed curvature is larger than a threshold, we regard this arc as an outlier. The numerical value of the curvature of the fitting residual graph depends on the scale of the horizontal axis of the residual graph. For example, if two arcs have the same shape and different scales, the curvature may have different values if the distance between consecutive points on the horizontal axis is regarded as the unit of length. Therefore, we describe the fitting residual of the αth point p_α of the arc under consideration by the vector

$$\boldsymbol{Q}_\alpha = (\frac{\lambda e_{\max}\alpha}{N}, e_\alpha)^\top, \tag{11}$$

where $e_{\max}$ is the maximum of the fitting residuals, N is the number of points of that arc, and λ is a constant for normalization[5]. The fitting residual e_α at point p_α is computed by

$$e_\alpha = \sqrt{\frac{(\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})}{(\boldsymbol{\theta}, V_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}}. \tag{12}$$

If $e_{\max}$ is extremely large, the normalization Eq. (11) may not work well. So, if $e_{\max}$ is larger than a threshold[6] $E_{\max}$, we replace the value $e_{\max}$ with the threshold $E_{\max}$ in the normalization computation.

For M segmented arcs $\boldsymbol{R}_\kappa(i,j)$ =$\{\boldsymbol{Q}_\beta | \beta = i, ..., j\}$,$\kappa = 1, ..., M$, we select inlier arcs by the following algorithm. Figure 3 summarizes the symbols used in the following algorithm.

1) Let α^* be the index of the point whose residual takes its maximum in the arc $\boldsymbol{R}_\kappa$. Here, if the residual at the point p_{α^*} is smaller than a threshold[7] $E_{\min}$, we regard the arc $\boldsymbol{R}_\kappa$ as a tangent arc and finish this procedure.

TABLE I
Comparison of computation time and number of iterations: Computation time in msec (number of iterations).

	Our method	RANSAC	Yu's method
(1)	8 (4)	54 (161)	632 (161)
(2)	8 (9)	42 (148)	600 (178)
(3)	4 (3)	64 (143)	424 (157)
(4)	8 (8)	68 (134)	48 (10)

2) Select two points whose indices β and γ are such that

$$\beta = \alpha^* - d, \ \gamma = \alpha^* + d, \ d = (j-i)/r, \tag{13}$$

where r is a constant[8] for determining the distance between the point α^* and its adjacent points β and γ. If both β and γ are out of $\boldsymbol{R}_\kappa$, we update d to $d \leftarrow d-1$ until either of the two points are in the arc $\boldsymbol{R}_\kappa$.

3) Compute two vectors $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ according to the following three rules. Here, $x_b^{(a)}$ denotes the b-th component of the vector $\boldsymbol{x}^{(a)}$.

Case a: Points $\boldsymbol{Q}_\beta$ and $\boldsymbol{Q}_\gamma$ are both in the arc $\boldsymbol{R}_\kappa$.

$$\boldsymbol{x}^{(1)} = \boldsymbol{Q}_{\alpha^*} - \boldsymbol{Q}_\beta, \ \boldsymbol{x}^{(2)} = \boldsymbol{Q}_{\alpha^*} - \boldsymbol{Q}_\gamma. \tag{14}$$

Case b: Point $\boldsymbol{Q}_\beta$ is in the arc $\boldsymbol{R}_\kappa$.

$$\boldsymbol{x}^{(1)} = \boldsymbol{Q}_{\alpha^*} - \boldsymbol{Q}_\beta, \ \boldsymbol{x}^{(2)} = (-x_1^{(1)}, x_2^{(1)})^\top. \tag{15}$$

Case c: Point $\boldsymbol{Q}_\gamma$ is in the arc $\boldsymbol{R}_\kappa$.

$$\boldsymbol{x}^{(2)} = \boldsymbol{Q}_{\alpha^*} - \boldsymbol{Q}_\gamma, \ \boldsymbol{x}^{(1)} = (-x_1^{(2)}, x_2^{(2)})^\top. \tag{16}$$

4) Compute the error curvature ϕ by

$$\phi = \pi - \cos^{-1}\left(\frac{(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})}{||\boldsymbol{x}^{(1)}||||\boldsymbol{x}^{(2)}||}\right). \tag{17}$$

5) Regard the arc $\boldsymbol{R}_\kappa$ as an inlier arc if ϕ is smaller than threshold $\hat{\phi}$.

We extend the selected inlier arc to its adjacent arcs to generate a longer arc in the first round of ellipse fitting. If the adjacent arc is a tangent arc, we test the next adjacent arc and regard the non-tangent arc as an inlier arc.

V. Experiment

A. Simulations

In order to confirm the effectiveness of our ellipse fitting method, we compared our method with RANSAC and Yu's method. Figure 4 shows the experimental results. The red ellipse of Fig. 4(a) is the ellipse fitted by our method and the blue points of Fig. 4(b) shows the selected inlier points for fitting the final ellipse (Fig. 4(a)). Figure 4(c), (d) and (e), (f) show the results of RANSAC and Yu's method, respectively, in the same manner as Fig. 4(a) and (b). We applied three methods to different variations of input data and Fig. 4(1), (2),

[5] We set λ=2.0.

[6] We set $E_{\max}$ be (maximum image coordinates of the input points)/3.

[7] We set $E_{\min} = 1.5$

[8] We set $r = 8$.

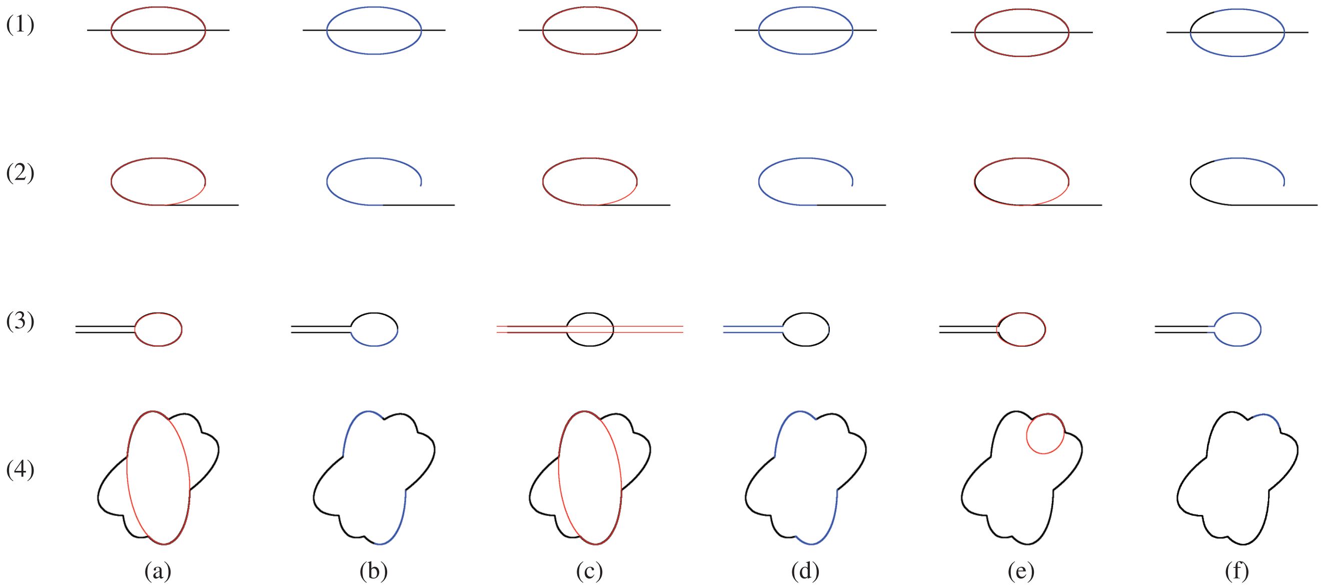

Fig. 4. Ellipse fitting results. (a), (c) and (e) The red ellipses are the fitted ellipse by our method, RANSAC, and Yu's method, respectively. (b) and (f) The blue points are the used points to fit those ellipse. (d) The blue points are the detected inliers by RANSAC.

(3) and (4) are four examples of them. As we can see that our method could fit correct ellipses even if outliers are contained inside of an elliptic point sequence shown in Fig. 4(1) and outliers are smoothly connected an elliptic arc point sequence shown in Fig. 4(2). RANSAC also fitted correct ellipses for Fig. 4(1),(2) and (4). However, degenerated conic, which is two lines, is fitted for Fig. 4(3). Yu's method fitted a slightly deviated ellipse from the correct one for Fig. 4(3) and a small ellipse to a short arc for Fig. 4(4).

Table I shows the number of iterations and computation times for three methods. We used Intel Core 2Duo 3.00GHz×2 for the CPU with main memory 4GB and Ubuntu 12.04 for the OS. For RANSAC, we stopped if the solution did not change after 50 consecutive iterations and counted the mean total number of iterations over 10 trials. From this result, the number of iterations and the computation time of our method is superior to RANSAC and Yu's method.

B. *Real image experiment*

We projected a circular cone of light onto a region where a floor and a wall meet and captured its camera image. Figures 5(a) show two input images. Figure 5(1)(a) is a very simple scene, while Fig. 5(2)(a) is rather complicated with an extra object near the floor-wall boundary. We first extracted edge point sequences using the Canny operator and selected a spatially connected segment that has the largest number of points. Figures 5(b) show the selected edge segments.

We projected an ellipse onto the floor and captured its image in advance. From that reference image, we extracted edge points and fitted an ellipse to them. Using this reference ellipse, we removed from the selected edge segment those points that belong to the reference ellipse (Figs. 5(c)). We fitted an ellipse to the remaining edge points by our proposed method and estimated a floor-wall boundary.

Figures 5(d) show the experimental results. The red arcs are the reference ellipse obtained in advance, and blue arcs are the ellipses fitted by our method using Figs. 5(c). As we can see, our method can fit an almost correct ellipse even when the input edge point sequence contains outliers that have complicated form.

For the two ellipses of Fig. 5(1)(d), we computed λ of Eq. (5) and obtained three real roots $\lambda = -0.331839$, -0.337805, and -3.15419. We computed the mean of the first two roots, and then computed the line using Eq. (7). For the two ellipses of Fig. 5(2)(d), we also computed λ of Eq. (5) and obtained three real roots $\lambda = -0.285334$, -0.296089, and -2.35997. We computed the mean of the first two solutions, and then compute the line using Eq. (7). The green lines of Figs. 5(d) indicate the computed lines. We can see that the floor-wall boundary is almost correctly estimated.

VI. Conclusion

In this paper, we proposed a new method for detecting the boundary between a floor and a wall by projecting a circular cone of light from a robot-mounted projector and extracting an ellipse in its image. Near the floor-wall boundary, two ellipses appear; we estimate the line that separate them, which we regard as the floor-wall boundary. We summarized our technique for analytically estimating the intersection line.

We also proposed a new method for fitting an ellipse to an edge point sequence which may contain outliers, i.e., non-elliptic arcs. Assuming that input points are a spatially connected segment, we segmented it to partial arcs based on

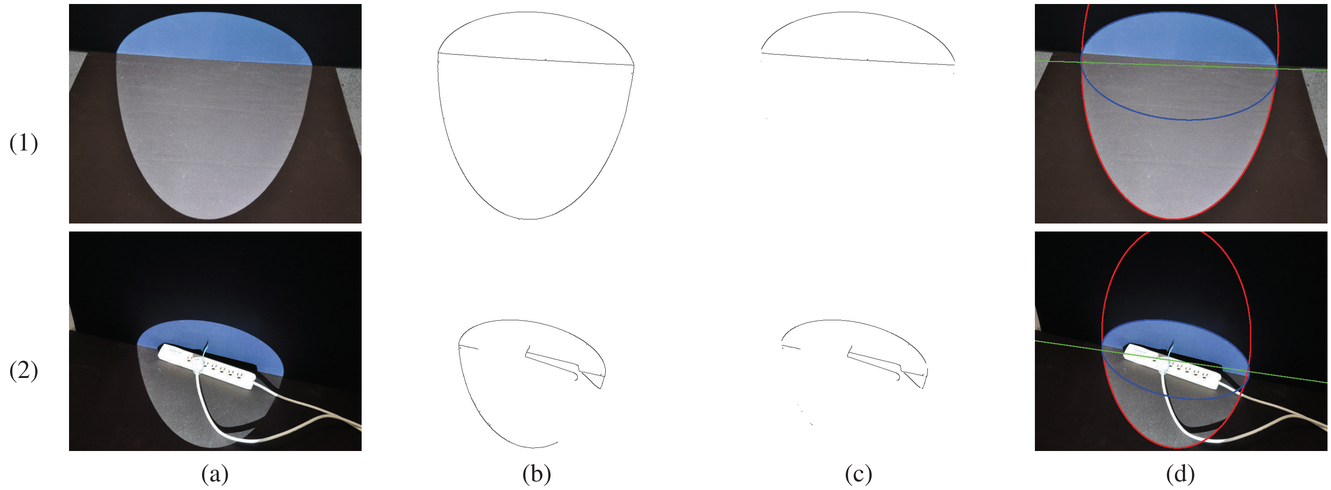

Fig. 5. The result of real image experiments. (a) Input images. (b) Selected edge segments. (c) Input edge points for our ellipse fitting method. (d) Fitted ellipses and estimated floor-wall boundaries. The red arcs are the reference ellipses fitted in advance. The blue arcs indicate the ellipses fitted by our method. The green lines are the estimated floor-wall boundaries.

the ellipse fitting residuals and detected inliers by computing the curvature of the residual graph for each divided arc.

By using simulated data, we compared the performance of our ellipse fitting method with RANSAC and Yu's method, and showed that the accuracy and computation time of our method are superior to existing methods. We also tested our proposed floor-wall boundary detection technique using real images. From the experimental results, we found that our method could accurately detect the floor-wall boundary by extracting an ellipse projected onto the scene.

In our future work, we need to validate the accuracy change for various camera and projector locations. We also need to estimate the real distance between the robot and the detected floor-wall boundary and evaluate its accuracy.

Acknowledgment

This work was supported in part by JSPS Grant-in-Aid for Scientific Research(C) (26330192).

References

[1] M. Barth, H. Ishiguro and S. Tsuji, Determining Robot Egomotion from Motion Parallax Observed by an Active Camera, Proc. 12th Int. Joint Conf. Artificial Intell., pp. 1247–1253, 1991.

[2] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley, On the fitting of surfaces to data with covariances, IEEE Trans. Patt. Anal. Mach. Intell., **22**-11, pp. 1294–1303, 2000.

[3] M. A. Fischler and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Comm. ACM, **24**-6, pp. 381–395, 1981.

[4] A. Fitzgibbon, M. Pilu, and R. B. Direct least squares fitting of ellipses, *IEEE Trans. Patt. Anal. Mach. Intell.*, **21**-5 (1999-5), 476–480.

[5] K. Kanatani, Geometric Computation for Machine Vision, Oxford University Press, Oxford, U.K., 1993.

[6] K. Kanatani and N. Ohta, Automatic detection of circular objects by ellipse growing, *Int. J. Image Graphics*, Vol. 4, No. 1, pp. 35–50, 2004.

[7] K. Kanatani and Y. Sugaya, Performance evaluation of iterative geometric fitting algorithms, Comput. Stat. Data Anal., **52**-2, pp. 1208–1222, 2007.

[8] K. Kanatani, P. Rangarajan, Y. Sugaya and H. Niitsuma, HyperLS and its applications, IPSJ Trans. Comput. Vis. Appl., **3**, pp. 80–94, 2011.

[9] K. Kanatani, A. Al-Sharadqah, N. Chernov, and Y. Sugaya, Renormalization Returns: Hyper-renormalization and Its Applications, Proc. 12th Euro. Conf. Comput. Vis., **3**, pp. 385–398, 2012.

[10] Y. Leedan and P. Meer, Heteroscedastic regression in computer vision: Problems with bilinear constraint, Int. J. Comput. Vision, **37**-2, p. 127–150, 2000.

[11] T. Masuzaki, Y. Sugaya and K. Kanatani, High accuracy ellipse-specific fitting, 6th Pacific-Rim Symposium on Image and Video Technology (PSIVT 2013), pp. 314–324, 2013.

[12] D. Murray, and James J. Little, Using real-time stereo vision for mobile robot navigation, Autonomous Robots **8**-2, pp. 161–171, 2000.

[13] S. Oishi, Y. Jeong, R. Kurazume, Y. Iwashita, and T. Hasegawa, ND voxel localization using large-scale 3D environmental map and RGB-D camera, IEEE Int. Conf. on Robotics and Biometrics (ROBIO2013), pp. 538–545, 2013.

[14] K. Onoguchi, N. Takeda and M. Watanabe, Planar projection stereopsis method for road extraction, IEICE TRANSACTIONS on Information and System 81.9, pp. 1006–1018, 1998.

[15] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, Visual modeling with a hand-held camera, Int. Journal of Computer Vision, **59**-3, pp. 207-232, 2004.

[16] E. Prassler and J. Scholz, Tracking Multiple Moving Objects for Real-Time Navigation, Autonomous Robots, **8**-2, pp. 105–116, 2000.

[17] Y. Sugaya, Ellipse Detection by combining division and model selection based integration of edge points, 4th Pacific-Rim Symposium on Image and Video Technology(PSIVT), pp. 64–69, 2010.

[18] Z. L. Szpak, W. Chojnacki, and A. van den Hengel, Guaranteed ellipse fitting with Sampson distance, Proc. 12th Euro. Conf. Comput. Vis., **5**, pp. 87–100. 2012.

[19] Yu Qiao, and S. H. Ong, Arc-based evaluation and detection of ellipses, Journal of Pattern recognition, **40**-7, pp. 1990–2003, 2007.