Least-square Matching for Mobile Robot SLAM Based on Line-segment Model

Sang-Hyung Park and Soo-Yeong Yi*

International Journal of Control, Automation and Systems 17(11) (2019) 2961-2968

ISSN:1598-6446 (print version)
eISSN:2005-4092 (electronic version)

To link to this article:

International Journal of Control, Automation and Systems 17(11) (2019) 2961-2968
http://dx.doi.org/10.1007/s12555-018-9070-8
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Least-square Matching for Mobile Robot SLAM Based on Line-segment Model

Sang-Hyung Park and Soo-Yeong Yi*

Abstract: This study proposes an efficient simultaneous localization and mapping (SLAM) algorithm for a mobile robot. The proposed algorithm consists of line-segment feature extraction from a set of points measured by a LIDAR, association and matching between the line-segments and a map database for position estimation, and the registration of the line-segments into the map database for the incremental construction of the map database. The line-segment features help reduce the amount of data required for map representation. The matching algorithm for position estimation is efficient in computation owing to the use of a number of inliers as the weights in the least-squares method. Experiments are conducted to demonstrate the performance of the proposed SLAM algorithm, and the results show that the proposed algorithm is effective in the map representation and the localization of a mobile robot.

Keywords: Line-segment, localization, map-making, mobile robot, SLAM.

1. INTRODUCTION

Localization capability is essential for the autonomous navigation of a mobile robot. The widely used dead-reckoning method accumulates localization error and the error eventually becomes very large with time [1, 2]. A ranging sensor is helpful for localization by creating a local object map surrounding the mobile robot and matching it with a given global map [3, 4]. However, the global map of an unknown environment is burdensome to prepare in advance. Simultaneous localization and mapping (SLAM) is a method of localization for mobile robots involving the incremental and simultaneous construction of the global map from a local map. The burdensome preliminary global map can be avoided using SLAM [5, 6].

As the ranging sensor, a 2-dimensional light detection and ranging (LIDAR) system is widely used for SLAM of a mobile robot on 2-dimensional ground [7]. A point measured by LIDAR is represented by (x, y) in the moving coordinates of a mobile robot. A stereo imaging camera has been used for the ranging sensor [8, 9], and the RGB-D sensor has been adopted to implement SLAM in recent years [10]. In general, these sensors require more computational effort to extract the features of the environment of the mobile robot compared to 2-dimensinal LIDAR.

SLAM algorithms are grouped according to 1) the map representation method and 2) the position estimation method. Because the number of points measured by a ranging sensor increases rapidly with time, it is impossible to use the raw data from the sensor to represent the environment map. Moreover, it is difficult to determine the corresponding points between the point sets obtained at successive time steps. Therefore, it is common to extract geometric features such as a vertex, line, or curve from a set of measurement points at each time step and represent the map with those features to reduce the amount of map data and to improve the computational efficiency. To utilize existing image processing techniques for feature extraction from a set of points measured by LIDAR, rasterization was proposed to convert the point set into a 2-dimensional grid image in [11]. Pedraza et al. used the B-Spline curve to represent the map of the unstructured natural environment [12], and Nunez et al. proposed a natural feature extraction method from the laser scan using the curvature that characterizes line-segments, corners and curve segments from the laser scan [13]. In [14], the generalized surface tensor was proposed to extract the features according to the normal vectors of two adjacent points in the measurement data set. The surface tensor helps to extract various features such as vertices, lines, and curves.

The line features are useful to model the environment of mobile robot because they exist in the majority of indoor environment and are efficient in terms of the amount of

Manuscript received February 2, 2018; revised October 27, 2018 and March 13, 2019; accepted July 12, 2019. Recommended by Associate Editor Hyun Myung under the direction of Editor-in-chief Keum-Shik Hong. This research was supported by the research program funded by National Research Foundation (Ministry of Education) of Korea (NRF-2018R1D1A1B07044841).

Sang-Hyung Park and Soo-Yeong Yi are with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, 232, Gongneung-ro, Nowon-gu, Seoul, Korea (e-mails: scrtrees@gmail.com, suylee@seoultech.ac.kr).
* Corresponding author.

data and computational cost. In [15], straight lines were extracted from a set of measurement points and the environment was represented by the straight-line features. However, some straight-line features from different objects of environment are indistinguishable because they were described only by the distance from the origin and the slope angle. Thus in this study, a straight-line feature is split into some line-segments and the line-segment features are used to model the environment objects instead of the line features.

The extended Kalman filter (EKF) is the most frequently used algorithm for the position estimation of a mobile robot [16, 17]. EKF-SLAM was proposed for simultaneous position estimation and map creation in [18, 19]. The state vector of EKF-SLAM includes the features of the map representation as well as the position of a mobile robot, and the computational efficiency of EKF-SLAM deteriorates with the number of features. To improve the computational efficiency of EKF-SLAM, FASTSLAM, which excludes features having a low association index through a particle filter algorithm, has been proposed [20, 21].

The main aim of this study is to present an efficient SLAM algorithm for an indoor mobile robot. The SLAM algorithm in this study consists of two parts: the incremental construction of an environment map database using the line-segment features and position estimation based on the simple least-squares method. The line-segment features are extracted from a set of points measured by LIDAR; in this study, a line-segment is described by two end points and the number of inlier points. An inlier point is a point measured by a ranging sensor that belongs to a line-segment. The number of inlier points represents the weight of each line-segment, which helps to reduce the error of matching between a line-segment and the map database for position estimation.

The remainder of this study is organized as follows: Section 2 describes the proposed SLAM algorithm including the line-segment extraction, association of a line-segment with the map database, matching for position estimation, and the registration of a line-segment in the map database. The results of experiments to verify the proposed algorithm and concluding remarks are presented in Section 3 and Section 4, respectively.

2. SLAM BASED ON LINE-SEGMENTS

Fig. 1 shows a flowchart of the SLAM algorithm used in this study. The algorithm involves 1) the extraction of line- segments from a set of points measured by LIDAR, 2) matching with the present map database and the localization of the mobile robot, and 3) updating the map database by the registration of the line-segments.

The variables used in this study are summarized as follows:

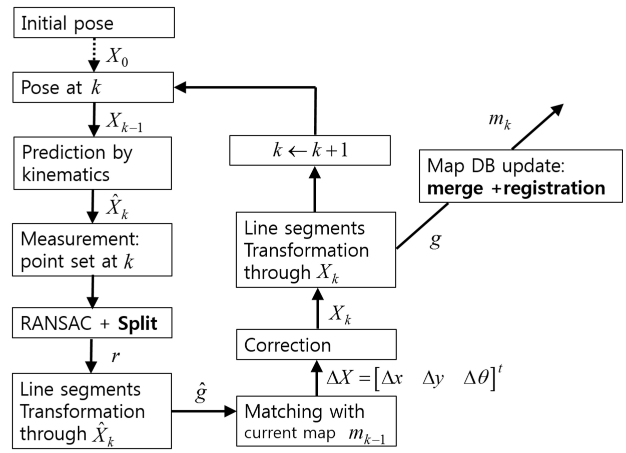
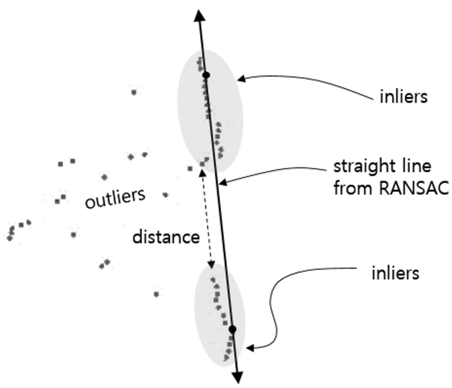


Fig. 1. Proposed SLAM algorithm.

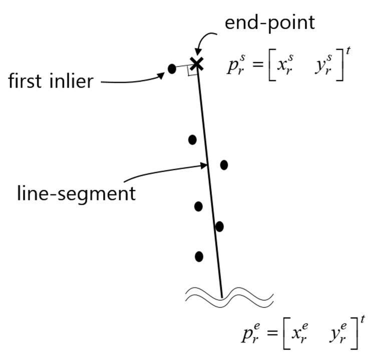
- $X_k = \left[x_k \quad y_k \quad \theta_k\right]^t$: The pose vector of the mobile robot including the position and heading angle in the global coordinate system. The subscript k denotes the time index.
- $\hat{X}_k = \left[\hat{x}_k \quad \hat{y}_k \quad \hat{\theta}_k\right]^t$: The prediction of the pose vector obtained by the kinematic model of the mobile robot.
- r: A line-segment extracted from a set of points measured by LIDAR in the moving coordinates of the mobile robot.
- n_r: The number of measured inlier points that belong to a line-segment r.
- $\hat{g}$: Transformation of the line-segment r into the global coordinates according to $\hat{X}_k$.
- m_{k-1}: A line-segment in the map database up to $k-1$.

2.1. Extraction of line-segments

To extract line-segments from a set of points obtained by a LIDAR sensor, the well-known random sample consensus (RANSAC) [22] algorithm is used in this study because of its computational efficiency. Because the original RANSAC algorithm extracts infinite straight lines rather than line-segments, this study proposes a simple split algorithm to obtain line-segments from the straight lines. By using the indices of the points measured by LIDAR, the split algorithm finds two consecutively indexed points whose distance exceeds a threshold value and divides the inliers into two clusters for each line-segment. Fig. 2(a) shows a straight line with the inliers extracted by the original RANSAC. The two end points of a line-segment are determined by the feet of perpendiculars of the first and the last indexed inliers, as shown in Fig. 2(b). The line-segment features are represented herein by two end points and the number of inlier points. The number of inlier points are used as the weighing factors in the least square matching algorithm in the later section.



(a) Classification for inliers into two clusters.



(b) End points of a line-segment.

Fig. 2. Extraction of line-segments from a straight line.

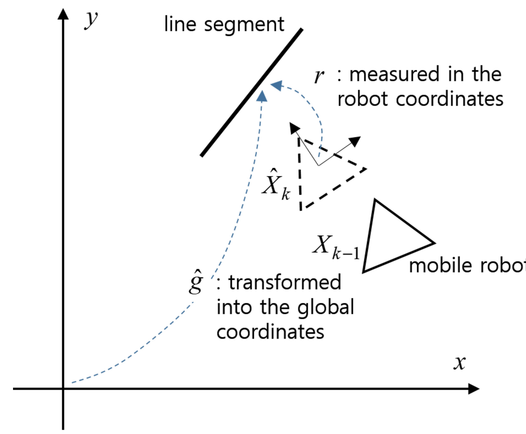


Fig. 3. Transformation from robot coordinates to global coordinates. The bold triangle denotes the pose of a mobile robot at $k-1$, the dotted triangle represents the prediction of the pose at k that is obtained by using the kinematic model (1) of the mobile robot, r implies a line-segment measured in the moving coordinates of a mobile robot, and $\hat{g}$ is the transformation of the corresponding line-segment into the global coordinates.

The two end points describing a line-segment are $p_r^s = \begin{bmatrix} x_r^s & y_r^s \end{bmatrix}^t$ and $p_r^e = \begin{bmatrix} x_r^e & y_r^e \end{bmatrix}^t$, and the number of inlier points that belong to the line-segment is n_r.

2.2. Association of line-segments

By matching the line-segments at k with the map database up to $k-1$, it is possible to obtain the amount of motion between the time interval and the pose vector of the mobile robot in global coordinates. Because a line-segment, r, at k from a LIDAR sensor is represented in the moving coordinates of the mobile robot, it should be transformed into global coordinates to associate it with the global map database. This transformation requires the pose information of the mobile robot in global coordinates. A possible pose vector of the mobile robot at k is obtained by prediction using the kinematic equation of motion with the known X_{k-1} as follows:

$$\begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \Delta t \begin{bmatrix} \cos\theta_{k-1} & 0 \\ \sin\theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{k-1} \\ w_{k-1} \end{bmatrix}, \tag{1}$$

where Δt denotes the sampling interval. Further, v_{k-1} and w_{k-1} are the input values. The transformation in (1) is depicted in Fig. 3. By using the predicted pose vector, $\hat{X}_k$, it is possible to transform two end points, p_r^s and p_r^e, of r in moving coordinates into global coordinates as follows:

$$\begin{bmatrix} x_{\hat{g}} \\ y_{\hat{g}} \end{bmatrix} = \begin{bmatrix} \cos\hat{\theta}_k & -\sin\hat{\theta}_k \\ \sin\hat{\theta}_k & \cos\hat{\theta}_k \end{bmatrix} \cdot \begin{bmatrix} x_r \\ y_r \end{bmatrix} + \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix}, \tag{2}$$

where $\hat{x}_k$, $\hat{y}_k$, and $\hat{\theta}_k$ are components of $\hat{X}_k$; $\begin{bmatrix} x_r & y_r \end{bmatrix}^t$ denotes an end point p_r^s or p_r^e of r; and $\begin{bmatrix} x_{\hat{g}} & y_{\hat{g}} \end{bmatrix}^t$ represents the corresponding point, $p_{\hat{g}}^s$ or $p_{\hat{g}}^e$, in global coordinates.

Fig. 3. Transformation from robot coordinates to global coordinates. The bold triangle denotes the pose of a mobile robot at $k-1$, the dotted triangle represents the prediction of the pose at k that is obtained by using the kinematic model (1) of the mobile robot, r implies a line-segment measured in the moving coordinates of a mobile robot, and $\hat{g}$ is the transformation of the corresponding line-segment into the global coordinates.

A line-segment, $\hat{g}$ at k is associated with a line-segment, m_{k-1}, in the map database for matching on the basis of the shortest distance between them. The distance between the two line-segments is defined as the distance between the two cross points of the perpendicular bisector of $\hat{g}$, as shown in Fig. 4.

In case the perpendicular bisector of $\hat{g}$ does not have a cross point with any m_{k-1} or the distance between the cross points is greater than a threshold value, $\hat{g}$ is regarded as new information to be registered into the map database.

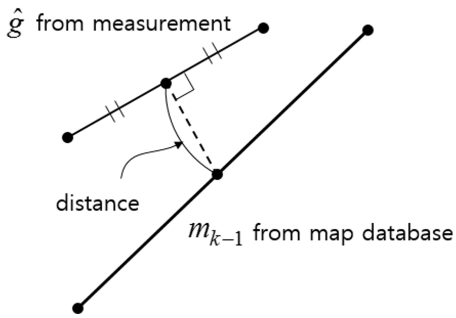


Fig. 4. Distance between two line-segments.

3. LEAST-SQUARE MATCHING FOR LINE-SEGMENTS

It is possible to obtain the amount of motion, $\Delta X = \left[\Delta x \;\; \Delta y \;\; \Delta\theta\right]^t$, of a mobile robot in a sampling interval by matching the pairs of the corresponding line-segments, $\hat{g}$ and m_{k-1}. Equation (3) expresses the transformation of an end point, $p_{\hat{g}}^s$, of $\hat{g}$ by $\Delta X = \left[\Delta x \;\; \Delta y \;\; \Delta\theta\right]^t$.

$$q_{\hat{g}}^s = R(\Delta\theta)\left(p_{\hat{g}}^s - \hat{c}_k\right) + \hat{c}_k + T(\Delta x,\, \Delta y), \tag{3}$$

where $R(\Delta\theta)$ and $T(\Delta x,\, \Delta y)$ represent rotation and the translation, respectively, and $\hat{c}_k = (\hat{x}_k,\, \hat{y}_k)$ denotes the position of the mobile robot predicted using (1). The same transformation as (3) for the other end point, $p_{\hat{g}}^e$, gives $q_{\hat{g}}^e$. The resultant $q_{\hat{g}}^s$ and $q_{\hat{g}}^e$ are the two end points of a line-segment containing information on the motion, $\Delta X = \left[\Delta x \;\; \Delta y \;\; \Delta\theta\right]^t$, at k. The matching error, $e_{\hat{g}}$, is defined as the sum of the squares of distance from the end points, $q_{\hat{g}}^s$ and $q_{\hat{g}}^e$, to the corresponding line-segment, m_{k-1}, in M_{k-1} as follows:

$$\begin{aligned}
e_{\hat{g}} &= n_{\hat{g}}\left\{\left(e_{\hat{g}}^s\right)^2 + \left(e_{\hat{g}}^e\right)^2\right\}\\
&= n_{\hat{g}}\left\{\left(q_{\hat{g}}^s\cdot\mathbf{u}_m - b_m\right)^2 + \left(q_{\hat{g}}^e\cdot\mathbf{u}_m - b_m\right)^2\right\}\\
&= n_{\hat{g}}\left[\left\{R(\Delta\theta)\left(p_{\hat{g}}^s - \hat{c}_k\right) + \hat{c}_k + T(\Delta x, \Delta y)\right\}\cdot\mathbf{u}_m - b_m\right]^2\\
&\quad + n_{\hat{g}}\left[\left\{R(\Delta\theta)\left(p_{\hat{g}}^e - \hat{c}_k\right) + \hat{c}_k + T(\Delta x, \Delta y)\right\}\cdot\mathbf{u}_m - b_m\right]^2,
\end{aligned} \tag{4}$$

where $n_{\hat{g}}$ is the number of inliers of a line-segment, $\hat{g}$, and is the same as n_r. Fig. 5 shows the definition of the matching error of $\hat{g}$ and m_{k-1}. In (4), m_{k-1} is represented by a line-equation, $p_m \cdot \mathbf{u}_m = b_m$, as shown in Fig. 5, where $\mathbf{u}_m$ is a unit normal of m_{k-1}.

The sum of all matching errors of the pairs of $\hat{g}$ obtained at k and m_{k-1} in the map database is expressed as follows:

$$E(\Delta x, \Delta y, \Delta\theta) = \sum_{\hat{g}} e_{\hat{g}}. \tag{5}$$

The rotation matrix, $R(\Delta\theta)$ in (4) is linearized as (6).

$$R(\Delta\theta) = \begin{bmatrix}\cos\Delta\theta & -\sin\Delta\theta\\ \sin\Delta\theta & \cos\Delta\theta\end{bmatrix} \approx \begin{bmatrix}1 & -\Delta\theta\\ \Delta\theta & 1\end{bmatrix}. \tag{6}$$

By taking derivatives of (5) with respect to Δx, Δy, and $\Delta\theta$, it is possible to obtain the least-square solution, (7) that minimizes the total matching error.

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta\theta\end{bmatrix} = \begin{bmatrix}A_{2\times2} & B_{2\times1}\\ C_{1\times2} & D_{1\times1}\end{bmatrix}^{-1}\begin{bmatrix}E_{2\times1}\\ F_{1\times1}\end{bmatrix}. \tag{7}$$

The components of (7) are as follows:

$$\begin{aligned}
A_{2\times2} &= \sum_{\hat{g}\in\hat{G}} 2n_{\hat{g}}\mathbf{u}_m\mathbf{u}_m^t,\\
B_{2\times1} &= \sum_{\hat{g}\in\hat{G}} n_{\hat{g}}\left[\left\{M\left(p_{\hat{g}}^s - \hat{c}_k\right) + M\left(p_{\hat{g}}^e - \hat{c}_k\right)\right\}\cdot\mathbf{u}_m\right]\mathbf{u}_m,\\
C_{2\times1} &= B_{2\times1}^t,\\
D_{1\times1} &= \sum_{\hat{g}\in\hat{G}} n_{\hat{g}}\left[\left\{M\left(p_{\hat{g}}^s - \hat{c}_k\right)\cdot\mathbf{u}_m\right\}^2 + \left\{M\left(p_{\hat{g}}^e - \hat{c}_k\right)\cdot\mathbf{u}_m\right\}^2\right],\\
E_{2\times1} &= \sum_{\hat{g}\in\hat{G}} n_{\hat{g}}\left\{\left(r_m - p_{\hat{g}}^s\cdot\mathbf{u}_m\right) + \left(r_m - p_{\hat{g}}^e\cdot\mathbf{u}_m\right)\right\}\mathbf{u}_m,\\
F_{1\times1} &\hat{=} \sum_{\hat{g}\in\hat{G}} n_{\hat{g}}\left[\left(r_m - p_{\hat{g}}^s\cdot\mathbf{u}_m\right)\left\{M\left(p_{\hat{g}}^s - \hat{c}_k\right)\cdot\mathbf{u}_m\right\}\right.\\
&\quad \left. + \left(r_m - p_{\hat{g}}^e\cdot u_m\right)\left\{M\left(p_{\hat{g}}^e - \hat{c}_k\right)\cdot u_m\right\}\right],
\end{aligned}$$

where M is the following matrix:

$$M = \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}. \tag{8}$$

By using the amount of motion, $\Delta X = \left[\Delta x \;\; \Delta y \;\; \Delta\theta\right]^t$, from (7), the corrected pose vector, X_k, at k is obtained as follows:

$$X_k = \hat{X}_k + \Delta X, \tag{9}$$

where $\hat{X}_k$ is the value of the pose vector predicted from the kinematic equation of motion given by (1). In case that the matrix inversion in (7) is singular at a certain k, the robot pose should not be updated at the time step.

The map database is updated simultaneously with the matching. Before updating the map database, all line-

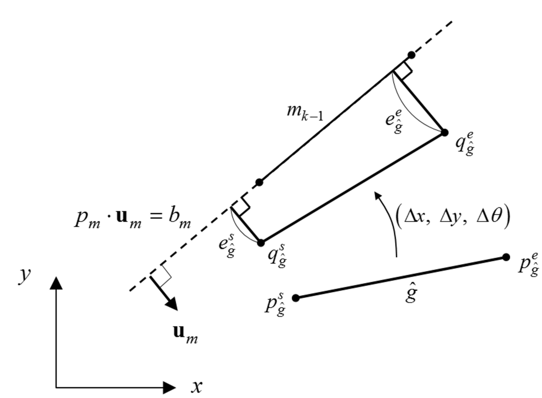


Fig. 5. Matching error of $\hat{g}$ and m_{k-1}.

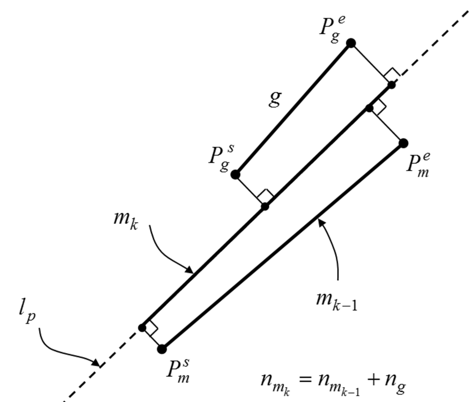


Fig. 6. Merging of two line-segments, g and m_{k-1}.

segments, r, at k should be transformed into global coordinates as follows according to the corrected pose vector, $X_k = \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^t$, of the mobile robot obtained by the matching:

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{bmatrix} \cdot \begin{bmatrix} x_r \\ y_r \end{bmatrix} + \begin{bmatrix} x_k \\ y_k \end{bmatrix}, \tag{10}$$

where $\begin{bmatrix} x_g & y_g \end{bmatrix}^t$ is an end point of g in global coordinates corresponding to r.

When a new line-segment is observed or an existing line-segment is changed at k, those are registered in the map database by updating the line-segment features. Two cases of the update are 1) adding a new line-segment, g, into the database that does not have any association in the map database and 2) merging a line-segment, g, with existing m_{k-1} in the map database if they are associated as described in Section 2.3. Fig. 6 shows a simple merging of two associated line-segments g and m_{k-1} in this study. In the figure, l_P denotes a straight line fitted by the least-square method for the four end points of g and m_{k-1}. The four end points have perpendicular feet onto l_P, and the longest pair of perpendicular feet determines two end points of the line-segment, m_k, to replace m_{k-1} in the map database. The number of inliers of m_k is simply $n_{m_k} = n_{m_{k-1}} + n_g$.

4. EXPERIMENTAL RESULTS

Fig. 7 shows the bi-wheel-type mobile robot used in this study to verify the proposed SLAM algorithm. Each wheel has an optical encoder to measure the input values, v_{k-1} and w_{k-1} in (1). The LIDAR sensor adopted in this experiment has a measurable distance and angle of 5.6 m $\times 240°$ [23]. The distance-measurement error of the LIDAR sensor is ± 30 mm, the angular resolution is $0.352°$, and the sampling interval is 0.1 s. Because of the offset distance, $d_s = 140$ mm, between the LIDAR and the center of the mobile robot, the data measured by the sensor are transformed first into the moving coordinates of the

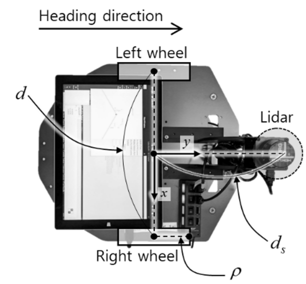
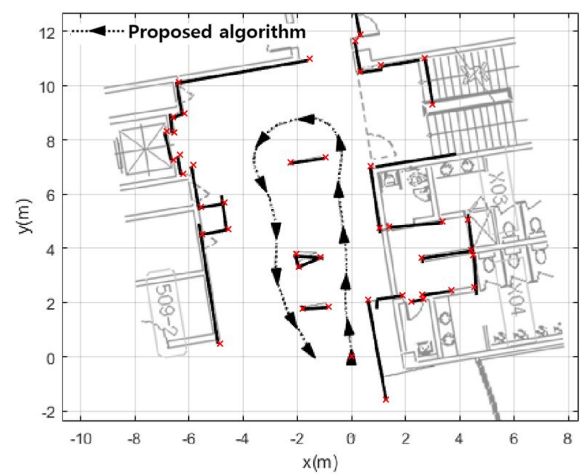


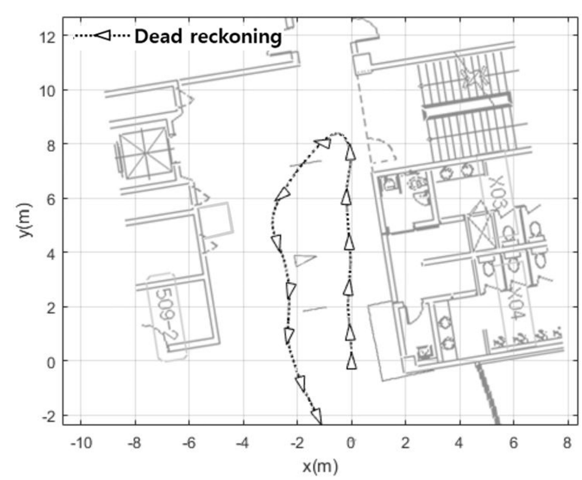
Fig. 7. Bi-wheel-type mobile robot used for the verification experiment.

robot.

Fig. 8 shows results of the actual experiments performed in a wide hall of a building. The bold line-segments in Fig. 8(a) represent the constructed map and



(a) Constructed map (bold line-segments) and mobile robot path estimated by the proposed algorithm.



(b) Mobile robot path estimated by dead-reckoning.

Fig. 8. Result of the experiment conducted in a wide hall of a building.

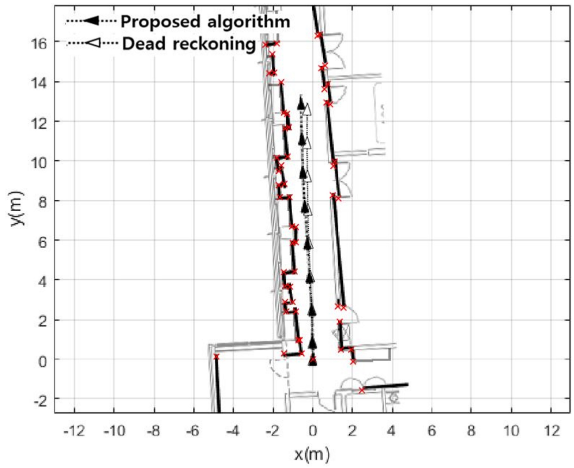


Fig. 9. Result of the experiment conducted in a narrow corridor of a building.

the dotted curve with arrows imply the path of a mobile robot estimated by the proposed SLAM algorithm. The map is superposed on a blueprint of the building for verification. For comparison, Fig. 8(b) shows the path of the mobile robot (the dotted curves with white arrows) estimated by dead-reckoning using encoder sensors at the two wheels without SLAM. The actual path of the mobile robot navigation in Fig. 8(a) and Fig. 8(b) is same. However, the estimated path in Fig. 8(b) is impractical because the mobile robot passes through an environmental object in the middle of the path.

Fig. 9 shows the result of another experiment conducted in a narrow corridor of a building. The bold line-segments in the figure represent the map of the building constructed along the path of the mobile robot by the proposed algorithm. The dotted curves with white arrows in the figure are the path of the mobile robot estimated by dead-reckoning for the same actual path. As shown, the dead-reckoning has error in the estimation of the actual path because it does not have any correction by using an external ranging sensor.

There are two kinds of methods for measuring the accuracy of SLAM algorithms: map accuracy measurement and localization accuracy measurement [24]. The localization accuracy measurement requires an additional equipment for global positioning of a mobile robot. Instead of the localization accuracy, the map accuracy is measured to show the performance of the SLAM algorithm in this study. Because the map is represented by a set of line-segments in this study, it is possible to quantify the accuracy of the constructed map by computing the error between the cross-points of a pair of the measured line-segments and the corresponding corner points in the blueprint of experimental environments. The points marked by ‘x’ in Fig. 8(a) and Fig. 9 are the points for the accuracy measurements. Table 1 shows the accuracy of the map constructed in the experiments.

Table 1. Accuracy of the constructed map.

	Distance error (m)
Experiment in Fig. 8(a)	0.067
Experiment in Fig. 9	0.234

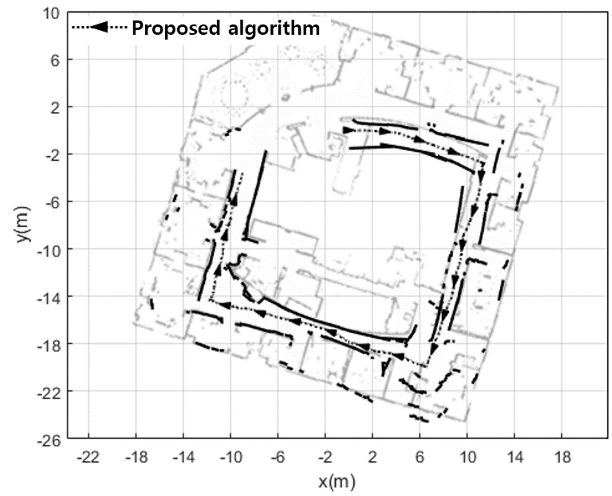


Fig. 10. Result of the proposed algorithm applied to an open dataset provided in [23].

The distance error in Table 1 is computed by

$$\text{Distance error} = \frac{\sum_{i=1}^{N} \sqrt{(x_{g,i}-x_{m,i})^2 + (y_{g,i}-y_{m,i})^2}}{N}, \tag{11}$$

where N is the number of the points and $(x_{m,i}, y_{m,i})$ and $(x_{g,i}, y_{g,i})$ are the coordinate values of the measured corner points and the corresponding ground-truth corner points from the blueprints of the experimental environments.

The proposed algorithm is applied to an open dataset provided in [25]. In order to avoid the loop-closure detection problem [26], part of a single session of the dataset is used in this study. The constructed map and the estimated path of a mobile robot are shown in Fig. 10 where the light gray is the map of a building [27] and the bold line-segments represents the map constructed in this study. The dotted curve with arrows is the estimated path of a mobile robot. In the constructed map, there are some errors caused by the unknown initial position that require additional optimization using the loop-closure detection with the multi-session mapping [24].

5. CONCLUDING REMARKS

An efficient SLAM algorithm for indoor mobile robots is proposed in this study. The SLAM algorithm includes the extraction of line-segment features from a set of point data measured by a LIDAR ranging sensor, the association and matching between the line-segment features and the map database for the localization of the mobile robot, and the incremental construction of the map database using the line-segment features. It is possible to represent the indoor environment approximately by the line-segments in general. In this study, a line-segment was described by two end points and the number of inliers from the set of measurement points. Because the line-segment model extracts the essential features from raw measurement point data, the construction and manipulation of the environment map database are computationally simple and efficient. The number of inliers plays the role of weights in the matching for localization by the least-squares method and helps to reduce the localization error. Experimental results validate the performance of the proposed SLAM algorithm in terms of localization and simultaneous mapmaking.

REFERENCES

[1] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," *Proc. of IEEE Int'l Conf. on Robotics and Automation*, Cincinnati, USA, May 1990.

[2] M. Sualeh and G. Kim, "Simultaneous localization and mapping in the epoch of semantics: a survey," *International Journal of Control, Automation and Systems*, vol. 17, no. 3, pp. 729-742, 2019.

[3] I. Cox, "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 2, pp. 193-204, 1991.

[4] I. Cox and J. Kruskal, "On the congruence of noisy images to line segment models," *Proc. of Int'l Conf. on Computer Vision*, pp. 252-258, 1988.

[5] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," *Proc. of IEEE/RSJ Int'l Workshop on Intelligent Robots and Systems (IROS'91)*, pp. 1442-1447, 1991.

[6] M. Jung and J. Song, "Efficient autonomous global localization for service robots using dual laser scanners and rotational motion," *International Journal of Control, Automation and Systems*, vol. 15, no. 2, pp. 743-751, 2017.

[7] S. Riisgard and M. Blas, *SLAM for Dummies: A Tutorial Approach to Simultaneous Localizing and Mapping*, MIT, 2004.

[8] A. Davison, I. Reid, and N. Molton, "MonoSLAM: real-time single camera SLAM," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052-1067, Apr. 2007.

[9] N. Karlsson, E. Bernardo, and J. Ostrowski, "The vSLAM algorithm for robust localization and mapping," *Proc. of IEEE Int'l Conf. on Robotics and Automation*, Barcelona, Spain, Jan. 2006.

[10] J. Sturm, N. Engelhard, and F. Endres, "A benchmark for the evaluation of RGB-D SLAM systems," *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Dec. 2012.

[11] Y. Li and E. Olson, "Extracting general-purpose features from LIDAR data," *Proc. of IEEE Int'l Conf. on Robotics and Automation, Anchorage*, AK, USA, May 2010.

[12] L. Pedraza, D. Rodriguez-Losada, F. Matia, G. Dissanayake, and J. Miro, "Extending the limits of feature-based SLAM with b-splines," *IEEE Trans. on Robotics*, vol. 25, no. 2, pp. 353-366, April 2009.

[13] P. Nunez, R. Vazquez-Martin, J. del Toro, A. Bandera, and F. Sandoval, "Feature extraction from laser scan data based on curvature estimation for mobile robotics," *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pp. 1167-1172, May 2006.

[14] Y. Li and E. Olson, "Structure tensors for general purpose LIDAR feature extraction," *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA)*, Shanghai, China, Aug. 2011.

[15] A. Garulli, A. Giannitrapani, A. Rossi, and A. Vicino, "Mobile robot SLAM for line-based environment representation," *Proc. of Decision and Control, and European Control Conference. CDC-ECC '05. 44th IEEE Conf.*, Seville, Spain, Dec. 2005.

[16] S. Yi and B. Choi, "Autonomous navigation of indoor mobile robots using a global ultrasonic system," *Robotica*, vol. 22, pp. 369-374, 2004.

[17] J. Leonard and H. Durrant-whyte, "Simultanenous map building and localization for an autonomous mobile robot," *Proc. of IEEE/RSJ Int'l Workshop on Intelligent Robots and Systems (IROS'91)*, pp. 1442-1447, 1991.

[18] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 99-108, June 2006.

[19] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108-117, Sept. 2006.

[20] M. Montelmerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: a factored solution to the simultaneous localization and mapping problem," *Proc. of the AAAI National Conf. on Artificial Intelligence*, USA, 2002.

[21] D. Hahnel, W. Burgard, and D. Fox, and S. Thrun, "An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, Oct. 2003.

[22] K. Derpanis, "Overview of the RANSAC algorithm," *Technical Report*, Computer Science, York University, 2010.

[23] https://www.hokuyo-aut.jp

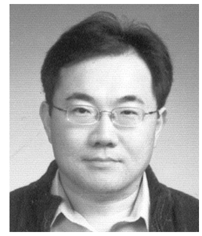
[24] R. Kummerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of SLAM algorithms," *Autonomous Robot*, vol. 27, pp. 387-407, 2009.

[25] http://ais.informatik.uni-freiburg.de/slamevaluation/datasets.php

[26] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular SLAM," *Robot. Auton. Syst.*, vol. 57, no. 12, pp. 1188-1197, 2009.

[27] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31-43, 2010.

Sang-Hyung Park received his B.S. and M.S. degrees in Electrical and Information Engineering from Seoul National University of Science and Technology, in 2016 and 2018, respectively. His research interests include nonlinear control for dynamic systems with unstable equilibrium point, and mobile robot.



Soo-Yeong Yi received his M.S. and Ph.D. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology, in 1990 and 1994, respectively. During 1995-1999, he stayed in Human Robot Research Center in Korea Institute of Science and Technology as a senior researcher. He was a professor in the Division of Electronics and Information Engineering, Chonbuk National University, Korea from September 1999 to February 2007. He was also a post doctorial researcher in the Department of Computer Science, University of Southern California, Los Angeles in 1997 and a visiting researcher in the Dept. of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign in 2005. He is now with the Department of Electrical and Information Engineering in Seoul National University of Technology, Korea. His primary research interest is in the area of robot vision, and intelligent control theory.