Robotica (2017) volume 35, pp. 1845–1863. © Cambridge University Press 2016
doi:10.1017/S0263574716000540

Robust 2D map building with motion-free ICP algorithm for mobile robot navigation

YoSeop Hwang and JangMyung Lee*

Department of Electronic Engineering, Pusan National University, Busan 609-735, Korea
E-mail: mmx001@pusan.ac.kr, jmlee@pusan.ac.kr

(Accepted July 9, 2016. First published online: August 15, 2016)

SUMMARY

A new motion-free iterative closest point (ICP) algorithm is proposed for building a two-dimensional (2D) map for mobile robot navigation. A laser range finder (LRF) sensor is installed on a mobile robot to scan and measure the depth data of the environment to form a 2D map during mobile robot navigation. Because the scanning and navigation motions are performed independently, the scanned data contain distortions from the motions of the mobile robot. To compensate for the distortions, the proposed motion-free ICP algorithm estimates the effects of the dynamic motions of the robot on the scanning process. That is, the motion-free algorithm compensates for the distance measurement errors related to the dynamic changes in the mobile robot's velocity. Experiments were performed with actual velocity changes of a mobile robot to demonstrate and verify the effective performance of the proposed algorithm.

KEYWORDS: 2D map building, Mobile robot, LRF, Navigation, Velocity compensation, ICP.

1. Introduction

A mobile robot must be able to move to a destination in order to carry out necessary tasks. To carry out operations successfully, the mobile robot must be able to locate its current position.[1–5] Recently, the social need for an aerial or mobile robot's localization has become widespread with enormous speed. As a consequence of these latest trends, newly developed mobile platforms that are adaptive and robust to unstructured environments have appeared. In an urban environment, mobile robots such as the Google self-driving car and Amazon drone have been demonstrated to perform tasks at a high level. In order to satisfy the social needs for mobile robots, this paper focuses on the development of high-speed localization techniques for projected two-dimensional (2D) spaces.

The mobile robot needs to recognize the surrounding environment with a sensor during its motion. By utilizing information on the environment, the mobile robot can create a map, or a graphic symbolic representation of significant features for part of the Earth's surface, for navigation. Based on the completed map of the environment, the robot can estimate its position, plan a path to follow, and manage autonomous driving. Depending on how the information on the motion and environment is saved, the map can be classified as an occupancy grid, feature-based, topological, or scan matching map.[6–13] When a 2D map is established, distance-estimation algorithms based on the extended Kalman filter or particle filter can be used. The estimation allows a map-matching algorithm to combine information on distances and remove duplicate parts prior to integration. Among the map-matching methods, the iterative closest point (ICP) algorithm is known to be the most accurate in combining and optimizing data.[14]

A mobile robot can move at a constant velocity while mapping or stop its motion before measuring the surrounding environment. However, such methods for mapping make it impossible to handle any changes in the velocity of the mobile robot. Naturally, the motion of the mobile robot produces errors. In this work, a 2D map was drawn by using environmental data measured with a laser range finder (LRF) sensor. Particular focus was given to developing compensations for the 2D map based

* Corresponding author. E-mail: jmlee@pusan.ac.kr

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press

on variable changes related to the driving velocity of the robot. Parameters that require careful consideration for processing the data of an LRF sensor attached to a mobile robot include the measurement time for the sensor data and the position of the robot at that moment.

The attached LRF should obtain distance information with the mobile robot at various speeds. Speed fluctuations obviously have a large impact on the LRF accuracy, which inevitably degrades the map-building performance. As the mobile robot moves faster, the tracking and map-building performance worsens because the mobile platform's speed fluctuation can be relatively large even though the scan rate of the LRF is constant. Thus, sensor calibration and LRF error compensation are essential processes for a high-speed test environment.

This paper introduces a new motion-free ICP algorithm to compensate for high-speed errors to enhance the 2D map-building performance. The algorithm was verified for two representative test environments: a long aisle in the shape of a square and an H-shaped aisle. The LRF sensor acquires environmental information by reading the distance data. In such a case, the data may be from the past because of the time delay when information is transmitted after raw data are processed. During a rotation, the scanning LRF sensor reads various kinds of data; there may be a time delay present in the measurement. Thus, there can be time differences during the interpretation of each dataset. In this work, the URG-04LX laser sensor was used; it can scan 240° and process distance measurement within 100 ms. Therefore, there can be a maximum time difference of 100 ms between the first and last data points. When the data provided by the sensor are read while the mobile robot is moving at a velocity of 1 m/s, there may be a maximum distance error of 0.1 m between the first and last data points. Thus, the error increases with the movement speed of the mobile robot.

To solve such problems, this paper suggests a motion-free algorithm to compensate for the velocity that is based on the ICP algorithm. The original ICP algorithm does not account for the scale factor in the least squares (LS) problem.[15,16] In practice, however, the scale factor exists universally in registration because images acquired by real digitizers differ greatly in viewpoint, scanning resolution, etc. In order to solve this kind of problem, the ICP algorithm can be combined with a model of the mobile robot; this is referred to as the "motion-free ICP algorithm" in this paper.

This paper comprises six sections. Section 2 states a general LS problem and briefly reviews the ICP algorithm. Section 3 explains the modeling process of the mobile robot for obtaining accurate data when the mobile robot is in motion. The principle to compensate for the velocity of the mobile robot is described. Section 4 explains the conversion of the motion coordinates and the motion-free algorithm for drawing the 2D map. Section 5 presents a 2D map drawn by an actual mobile robot in experiments to verify the proposed method. Lastly, Section 6 summarizes the results and suggests some avenues of future research.

2. Problem Statement and ICP Algorithm

2.1. Problem formulation

In general, for the 2D map building by a moving robot, the ICP algorithm has been utilized. For the precise map building by ICP algorithm, a reliable reference dataset is required, which comes from the trajectory planning in general. However in the trajectory planning, the speed of the mobile robot has not been considered, which causes distance errors in the matching process of the ICP algorithm. In this research, the motion of the mobile robot has been estimated and utilized for modifying the reference dataset.

The registration of model data (m-D) point sets is a difficult problem. In this work, the point set registration was formulated as a general optimization problem.

Given two point sets in R^m, the model shape $M \in \{\mathbf{m}_i\}_{i=1}^{N_m}$ and data shape$D \in \{\mathbf{d}_i\}_{i=1}^{N_d}$, $(N_m, N_d \in N)$ can be assumed to have a large overlap. Two m-D point sets are matched by finding a transformation T with which D is registered to be in the best alignment with M. Thus, the formulation is based on the following LS problem:

$$e(R, \mathbf{t}) = \min_{j \in \{1,2,\ldots,N_m\}} \left(\sum_{i=1}^{N_d} \left\| \mathbf{m}_j - T\mathbf{d}_i \right\|^2 \right), \tag{1}$$

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press

where the error function $e(R, \mathbf{t})$ is defined by the transformation $T = \left[\begin{smallmatrix} R & \mathbf{t} \\ 0 & 1 \end{smallmatrix}\right]$ between two m-D point sets.

In this research, the transformation T has been re-evaluated against the robot motion, and a motion-free ICP algorithm has been proposed with this new reference data, T.

2.2. Motion-free ICP algorithm

The ICP algorithm proposed by Besl and Mckay[17] is an efficient method for handling rigid registration between two point sets.

Its objective is to find a rigid transformation with which D is registered to be in the best alignment with$M$. That is, $T = \left[\begin{smallmatrix} R & \mathbf{t} \\ 0 & 1 \end{smallmatrix}\right]$ of Eq. (1) comprises a rotation R and translation $\mathbf{t}$. The rigid registration between two point sets can be described in detail as follows:

$$e\left(R, \mathbf{t}\right) \;= \min_{T, J, t \in \{1,2,\ldots,N_m\}} \left(\sum_{i=1}^{N_d} \left\| \mathbf{m}_j - \left(R\mathbf{d}_i + \mathbf{t}\right) \right\|^2 \right), \tag{2}$$

where $\mathbb{R} \in \mathbb{R}^{3\times 3}$ is a rotation matrix, $R^{\mathbf{T}}R = \mathbf{I}$ (identity matrix), $\det(R) = 1$, and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector.

The ICP algorithm in Eq. (2) can achieve a good and fast matching between the model and measured datasets when there a precise model dataset is provided by the reliable reference dataset R and $\mathbf{t}$. The reference dataset can be obtained by the following two steps. First, the correspondence between the two point sets $\mathbf{m}_j$ and $\mathbf{d}_i$ is represented as

$$\mathbf{c}_k\left(i\right) \;= \arg\min_{j \in \{1,2,\ldots,N_m\}} \left(\left\| \mathbf{m}_j - \left(R_{k-1}\mathbf{d}_i + \mathbf{t}_{k-1}\right) \right\|^2 \right). \tag{3}$$

Second, a new transformation between two point sets $\{\mathbf{m}_{c_k(i)}\}_{i=1}^{N_d}$ and $\{R_{k-1}\mathbf{d}_i + \mathbf{t}_{k-1}\}_{i=1}^{N_d}$ is computed by minimizing the squared distance, which is represented as

$$\left(R^*, \mathbf{t}^*\right) \;= \arg\min_{R, \mathbf{t}} \left(\sum_{i=1}^{N_d} \left\| \mathbf{m}_{c_k(i)} - R\left(R_{k-1}\mathbf{d}_i + \mathbf{t}_{k-1}\right) + \mathbf{t} \right\|^2 \right). \tag{4}$$

Using Eqs. (3) and (4), R_k and $\mathbf{t}_k$ can be updated as

$$R_k = R^* R_{k-1} \tag{5}$$

$$\mathbf{t}_k = R^* \mathbf{t}_{k-1} + \mathbf{t}^*. \tag{6}$$

This new motion-free ICP algorithm is applied to the data shape and model shape obtained from the mobile robot, which is assumed to move at a constant velocity.

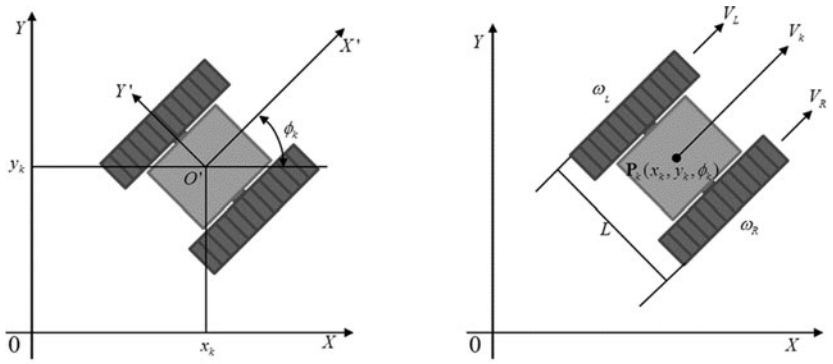
3. Velocity Compensation of LRF Sensor Based on Driving Velocity of Mobile Robot

If the velocity of a mobile robot changes while it is traveling on a straight or curved path, the measured data will have errors. More frequent changes in the velocity will increase the difficulty of setting the optimal model. Thus, compensation based on the change in velocity is required. In order for a mobile robot to measure an object accurately while changing velocity, it is necessary to first determine the velocity of the mobile robot and trace the motion because the position of the LRF sensor measuring the environmental information can differ depending on the velocity of the robot. In order to solve such a problem, the velocity and traced motion of the robot are calculated through a mechanical analysis of the mobile robot.

3.1. Modeling process of mobile robot

A mobile robot with a differential driving mechanism was used in this work. Based on the rotating angular velocity of both wheels, a mobile robot can move with two degrees of freedom. The current

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press



(a) Coordinates of mobile robot. (b) Kinematics of mobile robot.

Fig. 1. Mechanical analysis of mobile robot. (a) Coordinates of mobile robot. (b) Kinematics of mobile robot.

position of a mobile robot can be denoted as O'[18,19]:

$$O' = [x_k, y_k, \phi_k]^T. \tag{7}$$

Figure 1(a) shows the coordinates of a mobile robot; these are expressed as x_k, y_k, ϕ_k.

The relative coordinates of the robot are set so that the robot moves along the X'-axis at all times. The reference coordinates are given by OXY, whereas the positional coordinate of the robot is expressed by O'. The difference in the rotational angle between the relative coordinates $O'X'Y'$ and reference coordinates OXY of the robot is given by ϕ_k.

Figure 1(b) shows the angular velocities of the left and right wheels, which are measured with the encoder and are represented as ω_L and ω_R, respectively. The distance between both wheels is L, while the radius of each wheel is r which defines the pointing vector $\mathbf{r}$ from the center of the wheel to the contact point to the ground. The linear velocities of both wheels $\mathbf{v}_L$ and $\mathbf{v}_R$ can be calculated as follows:

$$\mathbf{v}_L = \boldsymbol{\omega}_L \times \mathbf{r} \tag{8}$$

$$\mathbf{v}_R = \boldsymbol{\omega}_R \times \mathbf{r}. \tag{9}$$

Now, the linear velocity of the robot $\mathbf{v}_k$ and the angular velocity Ω_k can be calculated as follows:

$$\mathbf{v}_k = \frac{\mathbf{v}_R + \mathbf{v}_L}{2} \tag{10}$$

$$\Omega_k = \tan^{-1}\left(\frac{\mathbf{v}_R - \mathbf{v}_L}{L}\right), \tag{11}$$

where the angular velocity is calculated in the counterclockwise direction.

When the kth position of the mobile robot is known, the $(k+1)$th position can be estimated as follows:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + |\mathbf{v}_k| \cdot \cos(\phi_k + \Omega_k \cdot c_T) \\ y_k + |\mathbf{v}_k| \cdot \sin(\phi_k + \Omega_k \cdot c_T) \\ \phi_k + \Omega_k \cdot c_T \end{bmatrix}, \tag{12}$$

where c_T is the control cycle of the mobile robot system.

Therefore, the nonlinear state equation of the dynamic model can be represented as follows:

$$\begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\phi}_k \end{bmatrix} = \begin{bmatrix} |\mathbf{v}_k| \cdot \cos(\phi_{k-1} + \Omega_k \cdot c_T) \\ |\mathbf{v}_k| \cdot \sin(\phi_{k-1} + \Omega_k \cdot c_T) \\ \Omega_k \cdot c_T \end{bmatrix}, \tag{13}$$

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press

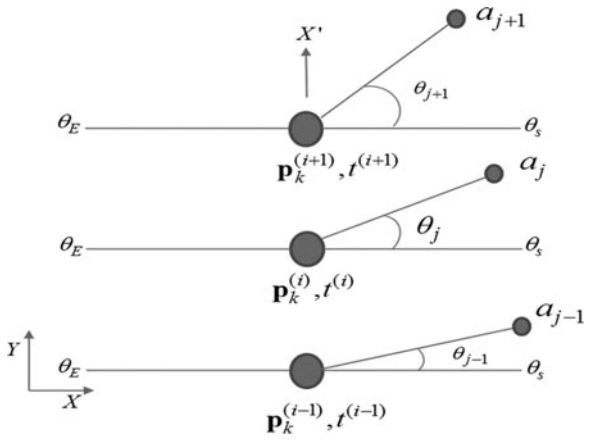


Fig. 2. Measurement position of LRF sensor based on motion of mobile robot.

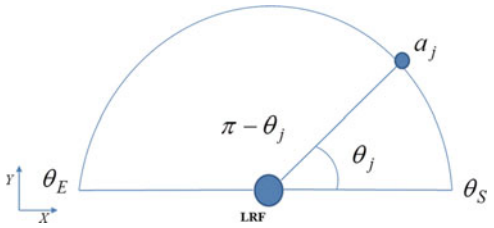


Fig. 3. Angular positions based on rotation of LRF sensor.

where $\mathbf{p}_k = [x_k, y_k, \phi_k]^{\mathrm{T}}$ can be defined as a state vector of the dynamic system, which represents the origin of the mobile robot in Fig. 1(b).

3.2. Implementation of motion-free ICP algorithm

Because the LRF sensor on the mobile robot rotates to measure the distance to the environment, the time differences of each measurement point need to be estimated for precise measurement of the distance. Because of these characteristics, the distance errors of the sensor increase when the mobile robot moves faster. In order to resolve this problem, the velocity at the measurement position of the sensor needs to be calculated.

Figure 2 shows the measurement position of the sensor when the mobile robot moves along the X' direction and the scanning sensor is located at the position $\mathbf{p}_k^{(i)}$ over the time $t^{(i)}$. In the figure, θ_S shows the starting position of the measurement, whereas θ_E represents the finishing position. When the mobile robot moves to the positions $\mathbf{p}_k^{(i-1)}$, $\mathbf{p}_k^{(i)}$, and $\mathbf{p}_k^{(i+1)}$, the measurement positions of the sensor become a_{j-1}, a_j, and a_{j+1}, respectively. Therefore, the measurement position of the LRF sensor depending on the motion of the mobile robot must be derived to obtain accurate environmental information.

Figure 3 represents the measurement range of the sensor as from θ_S to θ_E. The jth measurement angle of the LRF sensor θ_j can be calculated as

$$\theta_j = \theta_s + (\theta_E - \theta_S)\frac{j}{M-1}, \tag{14}$$

where M is the total number of scanning sensor data points for a cycle. Note that t_j needs to be calculated because the measured data depend on the time. The measurement time t_j can be obtained as follows:

$$t_j = t_s + \frac{\theta_j}{2\pi}T_s + t_{ND}, \tag{15}$$

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press

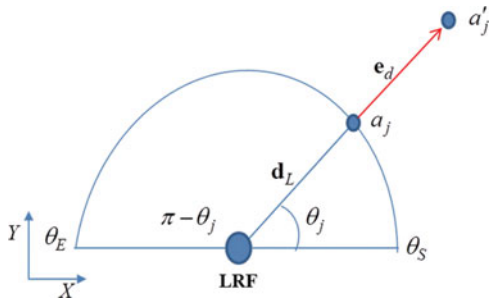


Fig. 4. The mobile robot speed is greater than V_R.

where t_s is the starting time of the scan, T_s represents the cycle time of the LRF sensor, and t_{ND} is the network delay time from the sensor.

The environmental information gathered by the LRF sensor for a cycle T_s from the scan start time t_s is represented as $D_t = \{d_j | j = 1, \ldots, M\}$, where d_j is the jth distance to the environment.

The absolute position of the environment measured by the scanner at the time t_j can be represented in terms of the robot position $\mathbf{p}_k$ and d_j. The absolute position of the environment at the time t_j can be obtained by interpolating the two points $(\mathbf{p}_k{}^{(i)}, t^{(i)})$ and $(\mathbf{p}_k{}^{(i+1)}, t^{(i+1)})$ as follows:

$$\mathbf{p}_k = \mathbf{p}_k + (t_j - t^{(i)}) \frac{\mathbf{p}_k{}^{(i+1)} - \mathbf{p}_k{}^{(i)}}{t^{(i+1)} - t^{(i)}}. \tag{16}$$

Finally, the position of an obstacle in the global coordinate system $({}^W x_j, {}^W y_j)$ at the time t_j can be represented byd_j and a_j based on $\mathbf{p}_k = [x_j, \ y_j, \ \phi_j]^{\mathrm{T}}$ as follows:

$${}^W x_j = x_j + d_j \cos(\phi_j + \angle a_j) \tag{17}$$

$${}^W y_j = y_j + d_j \sin(\phi_j + \angle a_j). \tag{18}$$

The distance error of the LRF sensor based on the velocity of the mobile robot is denoted as $\mathbf{e}_d$ and can be calculated as

$$\mathbf{e}_d = \mathbf{v}_k \cdot \frac{T_s}{M} \cdot n(D_t), \tag{19}$$

where $n(D_t)$ represents the number of scanned data points in a cycle.

By calculating the time at each position a_j with the known mobile robot velocity $\dot{\mathbf{p}}_k$, the distance of each position can be estimated. For this purpose, the safety velocity V_R of the mobile robot that does not go beyond the measuring range of the sensor is defined as follows:

$$V_R = \frac{\|\mathbf{d}_L\|}{T_s}. \tag{20}$$

where $\mathbf{d}_L$ is the maximum measurable distance vector of the LRF sensor.

The distance error $\mathbf{e}_d$ changes with the velocity of mobile robot. Based on the relationship between the safety velocity and mobile robot velocity, the distance error $\mathbf{e}_d$ can be calculated.

Figure 4 shows a situation where the velocity of the mobile robot $\mathbf{v}_k$ is greater than the safety velocity V_R.

When the velocity of the mobile robot is greater than V_R, the distance to a'_j becomes $\mathbf{d_L} + \mathbf{e}_d$. Then, the coordinates of the measured data $a'_j = (a'_{jx}, a'_{jy})$ can be calculated as follows:

$$a'_{jx} = \ ||\mathbf{d}_L + \mathbf{e}_d|| \cos\left(\theta_j\right) \tag{21}$$

$$a'_{jy} = \ ||\mathbf{d}_L + \mathbf{e}_d|| sin\left(\theta_j\right). \tag{22}$$

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press

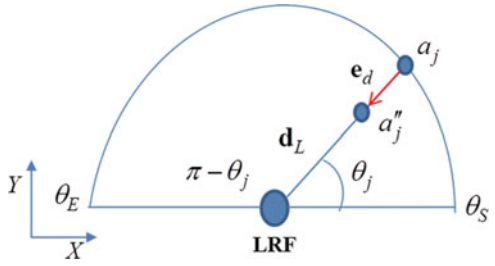


Fig. 5. The mobile robot speed is less than V_R.

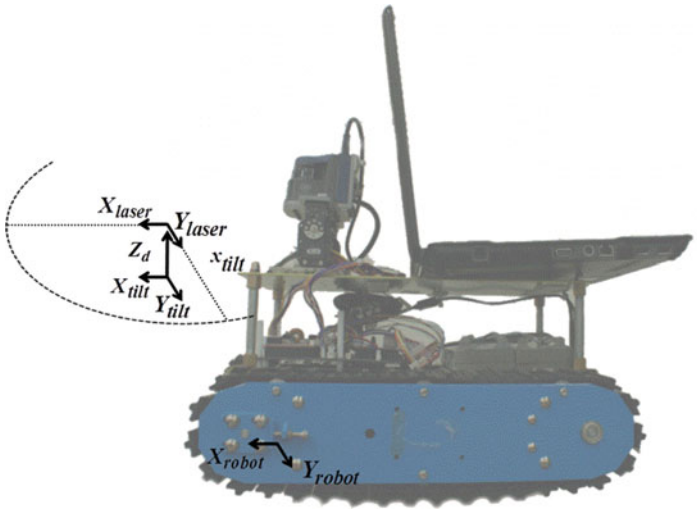


Fig. 6. Coordinates of the mobile robot.

Figure 5 shows the case when the velocity of the mobile robot $\mathbf{v}_k$ is less than the safety velocity V_R.

In this situation, the distance to a''_j becomes $\mathbf{d_L} - \mathbf{e}_d$, and the measured data $a''_j = (a''_{jx}, a''_{jy})$ can be obtained as follows:

$$a''_{jx} = ||\mathbf{d}_L - \mathbf{e}_d|| \cos(\theta_j) \tag{23}$$

$$a''_{jy} = ||\mathbf{d}_L - \mathbf{e}_d|| \sin(\theta_j). \tag{24}$$

By using these formulas, the effects of the mobile robot motion with a certain velocity $\mathbf{v}_k$ can be compensated for to realize precise measurement. That is, the measured location of the environment can be compensated from $a_j = (a_{jx}, a_{jy})$ to either $a'_j = (a'_{jx}, a'_{jy})$ or $a''_j = (a''_{jx}, a''_{jy})$ depending on the velocity of the mobile robot.

By using the compensated datasets of $a_j = (a_{jx}, a_{jy})$, error-free 2D data can be obtained as $\hat{M}$ (model set, $\hat{M} = \{\hat{\mathbf{m}}_i|_{i=1\ldots Nm}\}$) for the given velocities of $\mathbf{v}_k$, and $\hat{D}$ (measured dataset, $\hat{D} = \{\hat{\mathbf{d}}_j|_{j=1\ldots Nd}\}$) can be obtained in a single scan while the mobile robot is moving at an unknown velocity $\mathbf{v}_k$.

A new error function can thus be defined as follows:[20,21]

$$e(R, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left\| \hat{\mathbf{m}}_i - \left(R\hat{\mathbf{d}}_j + \mathbf{t} \right) \right\|^2, \tag{25}$$

where $w_{i,j}$ is 1 if $\hat{\mathbf{m}}_i$ is the same as $\hat{\mathbf{d}}_j$. Otherwise, $w_{i,j}$ is 0.

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press

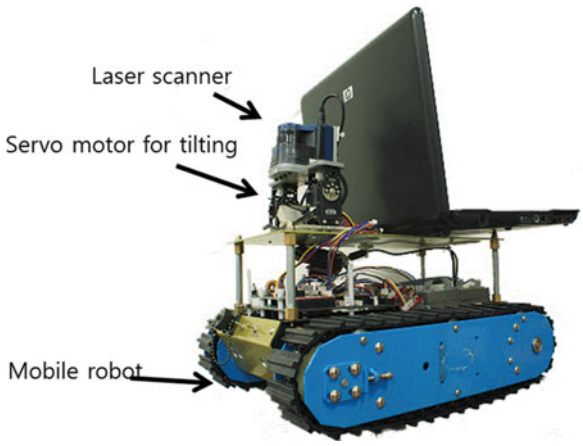


Fig. 7. Caterpillar-type mobile robot for experiments.

Therefore, the double summation in Eq. (25) can be simplified as follows:

$$e(R, \mathbf{t}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{\mathbf{m}}_i - \left(R\hat{\mathbf{d}}_i + \mathbf{t} \right) \right\|^2. \tag{26}$$

Instead of finding an optimal set of R and $\mathbf{t}$ for Eq. (26) concurrently, the optimal R is obtained first, and the optimal $\mathbf{t}$ is obtained later by using the optimal R. For this purpose, the average of the points used for the matching process can be defined as follows:

$$\mathbf{c}_m = \frac{1}{N} \sum_{i=1}^{N} \mathbf{m}_i \tag{27}$$

$$\mathbf{c}_\mathbf{d} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{d}_i. \tag{28}$$

Then, the augmented datasets are defined as follows:

$$M' = \left\{ \mathbf{m}'_i = \mathbf{m}_i - \mathbf{c}_m \right\}_{i=1,\ldots,N} \tag{29}$$

$$D' = \left\{ \mathbf{d}'_i = \mathbf{d}_i - \mathbf{c}_d \right\}_{i=1,\ldots,N}. \tag{30}$$

To obtain the optimal R from Eq. (26), the factor $\mathbf{t}$ is removed, and the augmented error function is represented with the augmented datasets $\hat{\mathbf{m}}'_i$ and $\hat{\mathbf{d}}'_i$ as follows:

$$\tilde{e}(R, \mathbf{t}) = \sum_{i=1}^{N} \left\| \hat{\mathbf{m}}'_i - R\hat{\mathbf{d}}'_i \right\|^2. \tag{31}$$

Because a rotation preserves the length, $\|R\mathbf{d}'_i\|^2 = \|\mathbf{d}'_i\|^2$ for the augmented error function in Eq. (31) is expanded as

$$\tilde{e}(R, \mathbf{t}) = \sum_{i=1}^{N} \left\| \mathbf{m}'_i \right\|^2 - 2 \sum_{i=1}^{N} \mathbf{m}'_i \cdot R\mathbf{d}'_i + \sum_{i=1}^{N} \left\| \mathbf{d}'_i \right\|^2. \tag{32}$$

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press

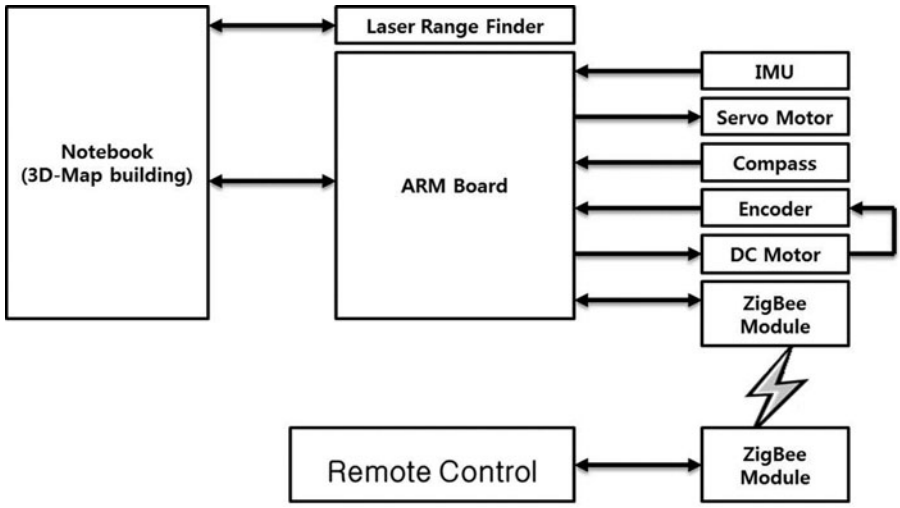


Fig. 8. Block diagram of overall system.

The rotation affects only the second term; thus, it is sufficient to maximize the second term, which can be calculated as $2\sum_{i=1}^{N}\mathbf{m}_i'^{\mathrm{T}} \times R\mathbf{d}_i'$. The second term can be represented by using the trace of a matrix as follows:[22]

$$\mathrm{tr}\left(\sum_{i=1}^{N} R\mathbf{d}_i' \times \mathbf{m}_i'^{\mathrm{T}}\right) = \mathrm{tr}(RH), \tag{33}$$

where

$$H = \sum_{i=1}^{N} \mathbf{d}_i' \times \mathbf{m}_i'^{\mathrm{T}}. \tag{34}$$

Now, the goal is to find a matrix R that maximizes $\mathrm{tr}(RH)$. By singular value decomposition, H is represented as follows:

$$H = \mathbf{U}\Lambda\mathbf{V}^{\mathrm{T}}, \tag{35}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal 2×2 matrices and Λ is a 2×2 diagonal matrix without negative elements.

A candidate optimal R is selected for the given H as follows:

$$R = \mathbf{V}\mathbf{U}^{\mathrm{T}}. \tag{36}$$

Then, from Eqs. (35) and (36)

$$\begin{aligned} RH &= \mathbf{V}\mathbf{U}^{\mathrm{T}}\mathbf{U}\Lambda\mathbf{V}^{\mathrm{T}}. \\ &= \mathbf{V}\Lambda\mathbf{V}^{\mathrm{T}} \end{aligned} \tag{37}$$

Note that the matrix RH is a symmetric, positive definite matrix. Therefore, for any given orthonormal matrix G, the following inequality condition is satisfied:

$$tr(RH) \geq tr(GRH). \tag{38}$$

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press

(a) Long and square-shaped experimental environment.

(b) H-shaped experimental environment.

Fig. 9. Real experimental environments. (a) Long and square-shaped experimental environment. (b) H-shaped experimental environment.

That is, the candidate $R(=\mathbf{V}\mathbf{U}^{\mathrm{T}})$ becomes a real optimal R.[23]

Now, the optimal translation is calculated by using the optimal R for a given H as follows:

$$\mathbf{t} = \mathbf{c}_m - R\mathbf{c}_d. \tag{39}$$

By Eqs. (36) and (39), the optimal rotation matrix and translation vector are obtained to finally match the two datasets $\hat{\mathbf{m}}_i'$ and $\hat{\mathbf{d}}_i'$.

4. Coordinate Transformation of Robot

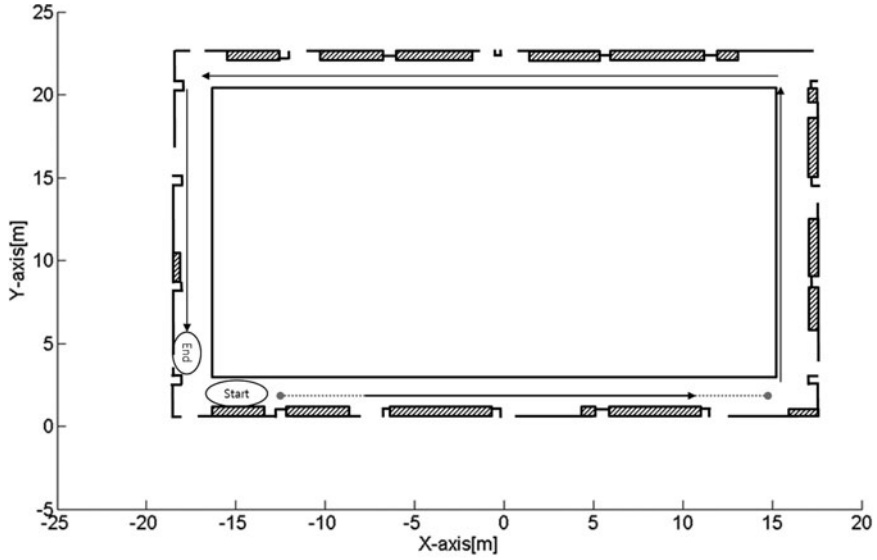
In order to draw the 2D map by using the laser scanner installed on the mobile robot, it is necessary to determine the coordinates for the position of the robot based on the global coordinates and the base of the LRF sensor w.r.t. the robot coordinates.

As indicated in Fig. 6, the coordinates of the robot (X_{robot}, Y_{robot}) and the coordinates of the laser (X_{laser}, Y_{laser}) w.r.t. the global frame can be represented as follows:

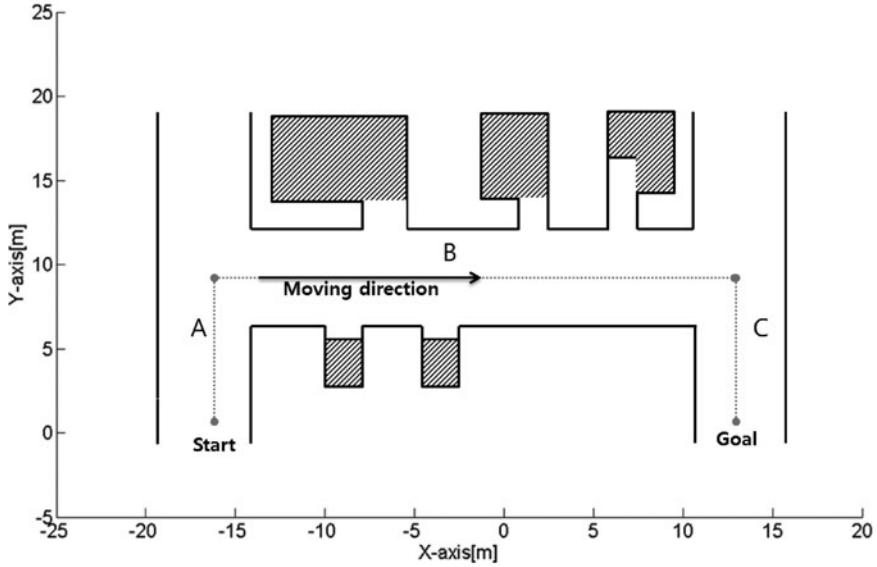
$$\begin{bmatrix} X_{robot} \\ Y_{robot} \end{bmatrix} = \begin{bmatrix} x_k + |\boldsymbol{v}_k| \times \cos(\phi_k + \Omega_k \cdot c_T) \\ y_k + |\boldsymbol{v}_k| \times \sin(\phi_k + \Omega_k \cdot c_T) \end{bmatrix} \tag{40}$$

where x_k and y_k represent the rectangular coordinates for the moving distance of the robot on the X- and Y-axes, respectively.

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press



(a) Model of long and square-shaped test environment.



(b) Modeling of H-shaped test environment.

Fig. 10. Model map of experimental environment. (a) Model of long and square-shaped test environment. (b) Modeling of H-shaped test environment.

Now, the laser sensor position w.r.t. the robot coordinates can be defined as

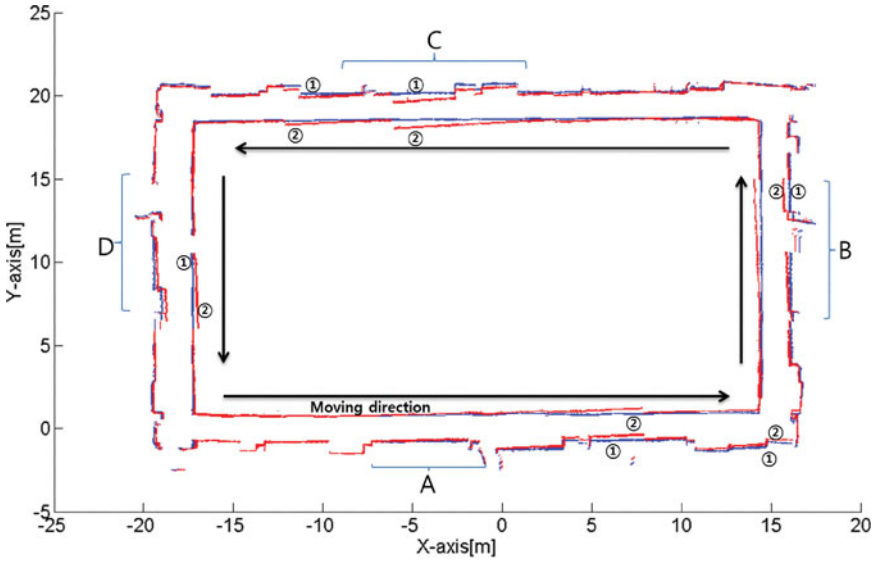
$$\begin{bmatrix} X_{Laser} \\ Y_{Laser} \end{bmatrix} = \begin{bmatrix} \cos(\theta_j) \\ \sin(\theta_j) \end{bmatrix} \times ||d_L||, \tag{41}$$

where θ_j represents the laser sensor angle (refer to Fig. 4) for the distance data measured by the LRF sensor $\mathbf{d}_L$.

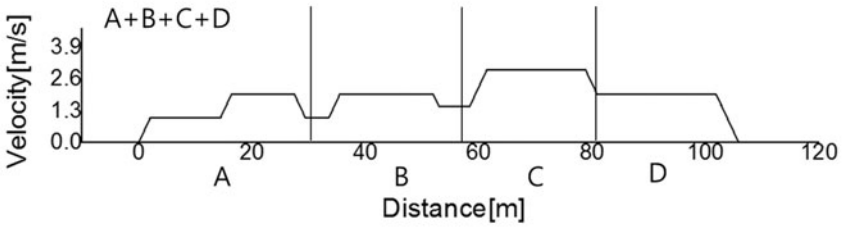
Combining the coordinates of the robot w.r.t. the global frame and the LRF sensor data w.r.t. the robot frame allows the 2D map data w.r.t. the global frame to be obtained as follows:

$$\begin{bmatrix} X_{map} \\ Y_{map} \\ 1 \end{bmatrix} = \begin{bmatrix} X_{Robot} \\ Y_{Robot} \\ 1 \end{bmatrix} + {}^{W}R_{R} \begin{bmatrix} X_{Laser} \\ Y_{Laser} \\ 1 \end{bmatrix} \tag{42}$$

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press



(a) Mapping result under squared aisle.



(b) Velocity changes of mobile robot under squared aisle.

Fig. 11. Velocities of the mobile robot along the corridor and the scanning results: (blue) stationary scan and (red) dynamic scan. (a) Mapping result under squared aisle. (b) Velocity changes of mobile robot under squared aisle.

where ${}^{W}R_{R}$ represents the rotational matrix defined by the relation between the robot frame and global frame.

5. Experimental Results

In order to verify the effectiveness of the proposed algorithm, a mobile robot equipped with a 10 W class DC motor and an optical encoder with a precision level of 256 pulses per rotation was used.

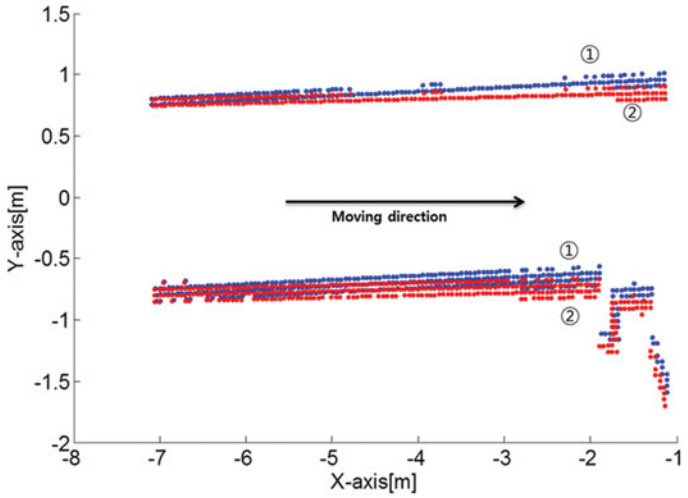
The URG-04LX model is an indoor-environment laser scanner from Hokuyo and is equipped with an RS-232 communication interface; it was used as the distance sensor. The 2D plane measuring range of the LRF sensor was 240° with intervals of 0.36°. The maximum measuring distance was 4000 mm. The roll, pitch, and yaw angles of the mobile robot were measured with an IMU sensor and a compass. Figure 7 shows the mobile robot used in the experiments.[24–29]

As shown in Fig. 8, the mobile robot was equipped with various sensors for position measurement. The main controller was designed with a PC. The moving parts can be controlled with two motors and motor drivers, which were used to control forward/backward movements with variable velocities. The ARM board controlled the mobile robot by receiving data provided by each sensor. The distance information was obtained by using a single laser distance meter.

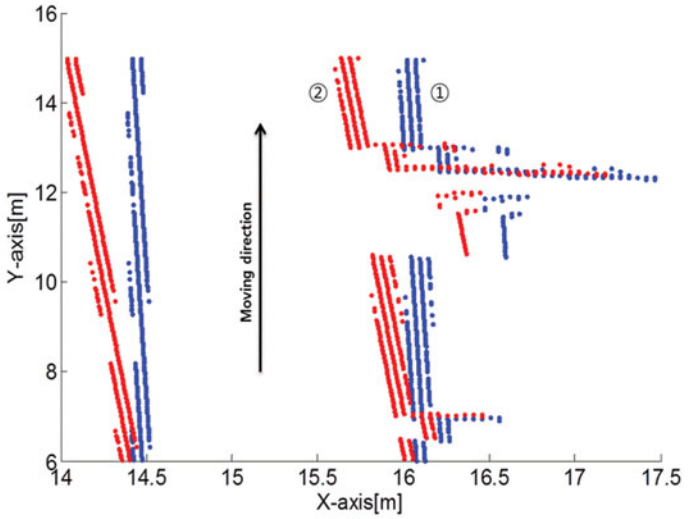
5.1. Experimental environment

The proposed algorithm was demonstrated for two different aisles. The first was long and square-shaped. In this environment, the experimental results could be used to verify the repetitive tracking

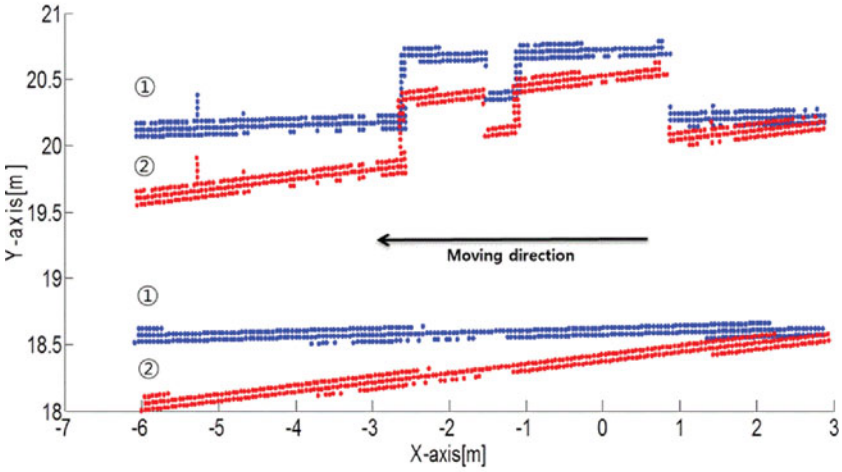
https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press



(a) Velocity = 1 m/s; first part of region A in Fig. 11



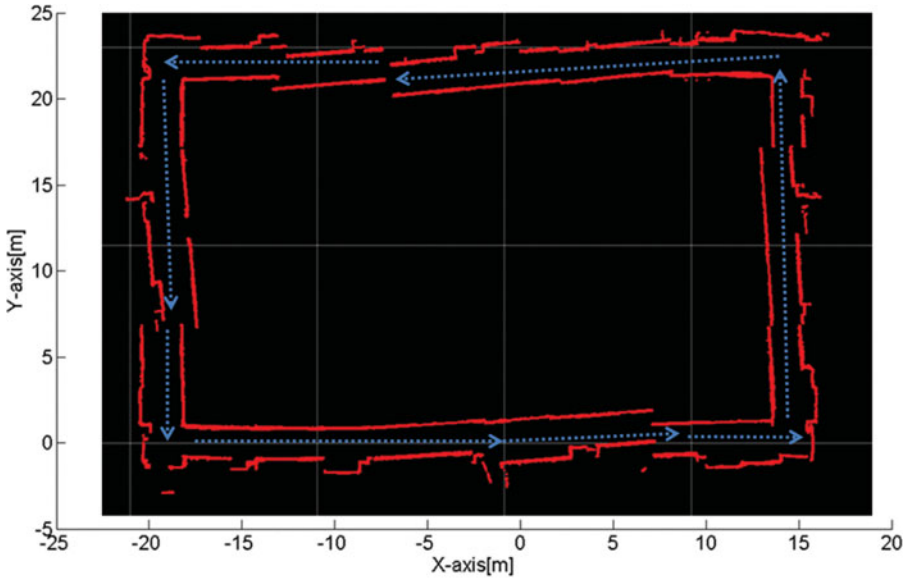
(b) Velocity = 2 m/s; region B in Fig. 11



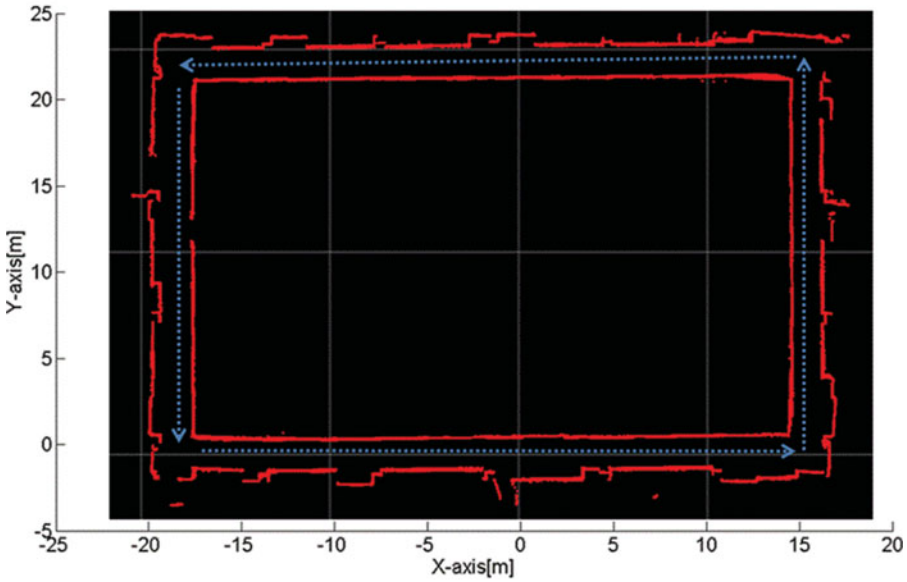
(c) Velocity = 3 m/s; middle of region C in Fig. 11

Fig. 12. Comparison of scanning errors according to the velocity: (blue) stationary scan and (red). (a) Velocity = 1 m/s; first part of region A in Fig. 11. (b) Velocity = 2 m/s; region B in Fig. 11. (c) Velocity = 3 m/s; middle of region C in Fig. 11.

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press



(a) Conventional ICP map for the square-shaped aisle.



(b) Motion-free ICP map for the square-shaped aisle.

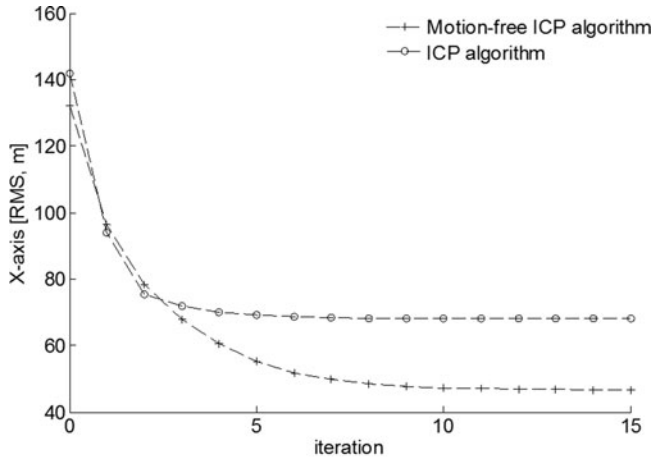
Fig. 13. Comparison of conventional and motion-free ICP maps. (a) Conventional ICP map for the square-shaped aisle. (b) Motion-free ICP map for the square-shaped aisle.

performance depending on the velocity fluctuation. The second was an H-shaped environment, where acceleration and deceleration are relatively hard to perform. This was used to evaluate the tracking performance according to rapid variations in the velocity of the mobile platform.

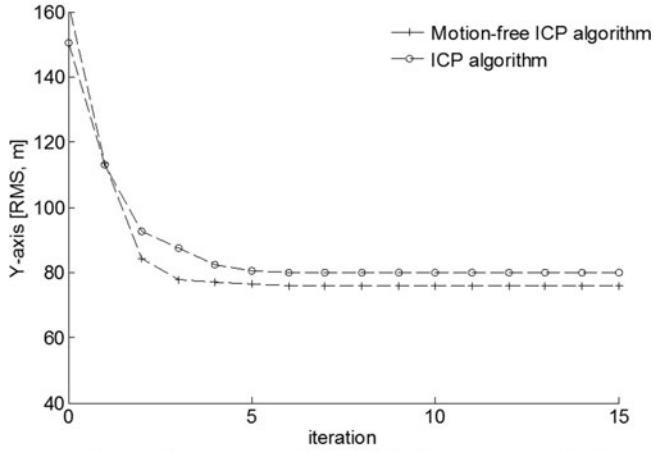
Figure 9(a) shows the long and square-shaped experimental environment, which was an indoor hallway. Figure 9(b) illustrates the H-shaped test environment. In order to determine the distance error depending on changes in the velocity, a corridor environment with walls on both sides was selected.

Figure 10 shows the experimental environment, where the robot moved along a straight line with changing velocities from the start to the goal. The route was divided into four sections with different mobile robot speeds.

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press



(a) Tracking error w.r.t. the x-axis for the square-shaped aisle.



(b) Scanning error w.r.t. the y-axis for the square-shaped aisle.

Fig. 14. Distance error for the square-shaped aisle environment. (a) Tracking error w.r.t. the X-axis for the square-shaped aisle. (b) Scanning error w.r.t. the Y-axis for the square-shaped aisle.

5.2. Experimental result 1: long and square-shaped environment

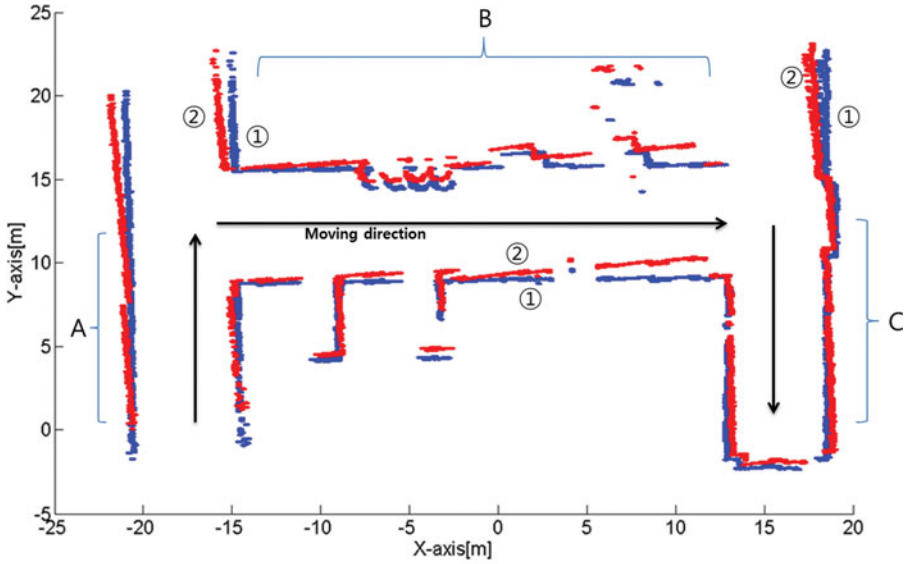
In Fig. 11(a), the blue line (1) represents the map data obtained by the stationary scan, whereas the red line (2) shows the map obtained by the dynamic scan. The latter was obtained while the mobile robot was moving at the given velocities. Even when there was no scanning error when the robot was stationary, the error started to increase when the velocity of the robot was increased to 1 m/s. The largest error was when the velocity became 3 m/s. Figure 11(b) shows the speed of mobile robot in four sections. The results showed that scanning errors occurred when the mobile robot moved faster while the scanning of the environment was kept the same.

Figure 12 shows the distance errors with different velocities of the mobile robot: (1) is the map obtained by the stationary scan, and (2) is the occurrence of distance errors depending on the motion of the mobile robot. The error increased with the velocity of the mobile robot.

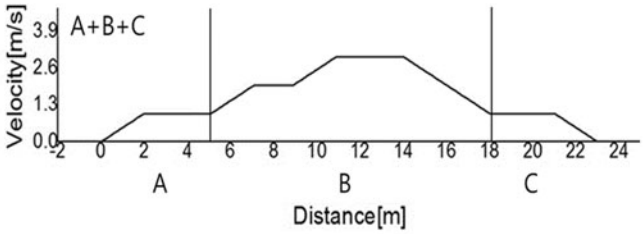
Figure 13(a) shows a conventional ICP map drawn as the mobile robot moved. Errors occurred because of the motion of the mobile robot. Figure 13(b) shows the map when the motion-free ICP algorithm was applied in order to correct the distance errors in the environmental map. The distance errors from the LRF measurement were confirmed to be removed by the motion-free ICP algorithm.

The computation times of the proposed motion-free ICP algorithm and conventional ICP algorithm were calculated for 15 iterations. The motion-free ICP algorithm was found to be more efficient; the average computation times of the conventional ICP and motion-free ICP were 645 and 525 ms, respectively.

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press



(a) Mapping result for the H-shaped aisle.



(b) Velocities at regions A, B, and C in Fig. 15 (a).

Fig. 15. Velocity changes of mobile robot and mapping result. (a) Mapping result for the H-shaped aisle. (b) Velocities at regions A, B, and C in (a).

Figure 14(a) shows the X-axis distance errors by the motion-free ICP and ICP algorithms, respectively. The ICP algorithm showed a slightly better improvement initially. However, it soon converged to a constant value and did not show any further improvement. The motion-free ICP algorithm showed a lower convergence speed than the ICP algorithm initially. However, it monotonically converged to a small value. The RMS (root mean square) value was used for an accurate comparison in case the distance errors were negative values.

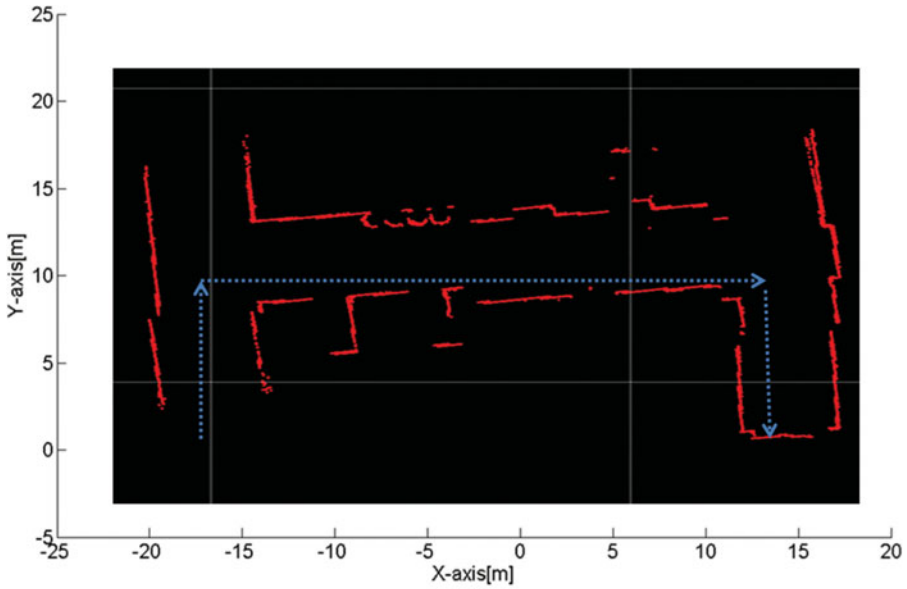
Because the mobile robot moved forward in the direction of the X-axis, the moving direction was usually parallel to continuous objects such as walls. Objects placed in front of the mobile robot were in the X-axis direction, unlike the ones placed on the sides of the mobile robot in the Y-axis direction during motion. Thus, the Y coordinate was constant in most cases. In this sense, the Y-axis required a shorter computation time than the X-axis.

Figure 14(b) shows the Y-axis distance errors of the motion-free ICP and ICP algorithms. The initial error for the ICP algorithm was smaller, but the errors with the motion-free ICP algorithm converged to a lower value as the number of iterations was increased.

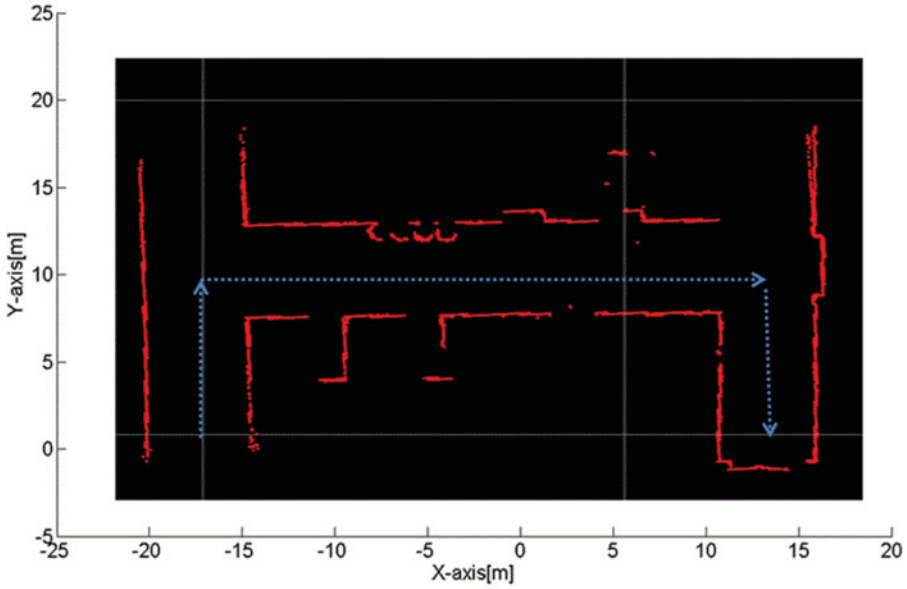
5.3. Experimental result 2: H-shaped environment

Figure 15(a) shows the mapping results for the H-shaped aisle environment. The blue line (1) represents the map data obtained by the stationary scan, whereas the red line (2) shows the map obtained by scanning from the moving mobile robot, which produced large distance errors. Figure 15(b) shows the speed of the mobile robot from region A to region C. The mobile platform accelerated to 1 m/s, turned right at the center point of the H shape, further accelerated to 3 m/s, and finally decelerated to 1 m/s.

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press



(a) Conventional ICP map for the H-shaped aisle.



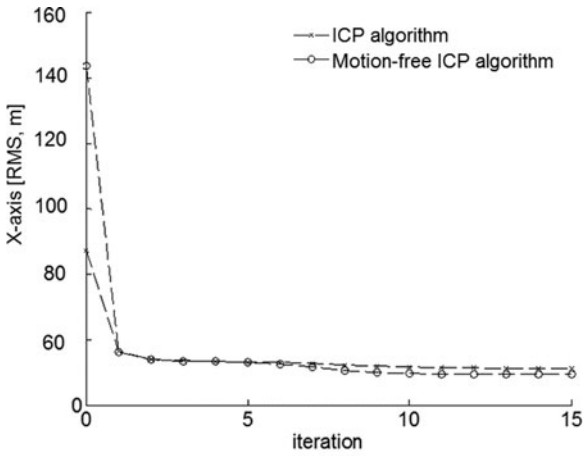
(b) Velocities at regions A, B, and C in Fig. 15 (a).

Fig. 16. Comparison of conventional and motion-free ICP maps. (a) Conventional ICP map for the H-shaped aisle. (b) Motion-free ICP map for the H-shaped aisle.

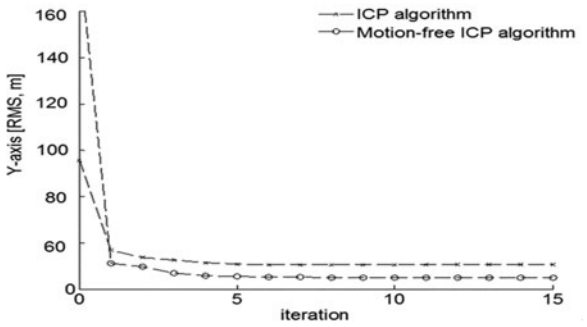
Figure 16(a) illustrates the conventional ICP map for the H-shaped aisle. A displacement error occurred because of the lack of velocity compensation for the mobile robot. In contrast, Fig. 16(b) shows the proposed motion-free ICP algorithm map for the same experimental environment. The motions of the mobile robot were compensated for by the motion-free ICP algorithm in real-time. For the experiments, the computation times of the conventional and proposed methods were evaluated for 15 iterations; the computation times of the two algorithms were 845 and 785 ms, respectively. Note that the scanning error increased around the edge of corner with the conventional ICP algorithm, which is an inevitable characteristic of using a mobile robot.

Figure 17 shows the X-axis distance errors with the motion-free ICP and ICP algorithms. The velocity fluctuation was more frequent in the H-shaped test environment than that in the square-shaped environment, so the degrees of the tracking error correspondingly decreased. Based on the

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press



(a) Tracking error w.r.t. the x-axis for the H-shaped aisle.



(b)Tracking error w.r.t. the y-axis for the H-shaped aisle.

Fig. 17. Distance error for the H-shaped environment. (a) Tracking error w.r.t. the X-axis for the H-shaped aisle.

above two environments, the proposed motion-free ICP algorithm provides a more stable and faster performance.

6. Conclusion

This paper presents a method of building a 2D map in an indoor environment that corrects errors based on changes in the velocity of the robot and the results of an experiment performed for verification. The motion-free ICP algorithm makes it possible to determine the occurrence of distance errors based on changes in the velocity of a mobile robot and the sensor characteristics of different experiments. This paper solves such problems with the motion-free ICP algorithm, which combines the ICP algorithm with compensation for the movement of a mobile robot. Actual experiments were performed with a mobile robot to demonstrate the effectiveness of the proposed algorithm. Future work with the motion-free ICP algorithm will involve drawing a 3D map by using an LRF on a fast-moving mobile robot. The estimation accuracy can be improved by the stochastic methods,[30,31] which are left as for future research.

Acknowledgement

This research was supported by the Ministry of Trade, Industry & Energy (MOTIE), Korea, under the Industry Convergence Liaison Robotics Creative Graduates Education Program supervised by KIAT (N0001126).

References

1. G. Allee, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots* **34**(3), 189–206 (2013).

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press

2. R. Kummerle, M. Ruhnke, B. Steder, C. Stachniss and W. Burgard, "A Navigation System for Robots Operating in Crowded Urban Environments," *Proceedings of IEEE Conference on Robotics and Automation ICRA*, (May 2013) pp. 3225–3232.
3. P. Henry, M. Krainin, E. Herbst, X. Ren and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," *Exp. Robot.*, Springer Berlin Heidelberg, 477–491 (2014).
4. J. Pacheco, J. Ascencio and J. Mancha, "Visual simultaneous localization and mapping: A survey," *Artif. Intell. Rev.* **43**(1), 55–81 (2015).
5. M. Mirkhania, R. Forsatib, A. Shahric and A. Moayedikiad, "A novel efficient algorithm for mobile robot localization," *Robot. Auton. Syst.* **61**(9), 920–931 (2013).
6. A. Censi, A. Franchi, L. Marchionni and G. Oriolo, "Simultaneous calibration of Odometry and sensor parameters for mobile robots," *Robotics* **29**(2), 475–492 (2013).
7. J. Fabian and G. M. Clayton, "Error analysis for visual odometry on indoor, wheeled mobile robots with 3-D sensors," *Mechatronics* **19**(6), 1896–1906 (2014).
8. R. Correala, G. Pajaresa and J. Ruzb, "Automatic expert system for 3D terrain reconstruction based on stereo vision and histogram matching," *Exp. Syst. Appl.* **41**(4), 2043–2051 (2014).
9. B. Dugarjav, S. Lee, D. Kim, J. Kim and N. Chong, "Scan matching online cell decomposition for coverage path planning in an unknown environment," *Int. J. Precis. Eng. Manuf.* **14**(9), 1551–1558 (2013).
10. H. Kwon1, K. Yousef and A. Kak, "Building 3D visual maps of interior space with a new hierarchical sensor fusion architecture," *Robot. Auton. Syst.* **61**(8), 749–767 (2013).
11. J. Bojja, M. Kirkko-Jaakkola, J. Collin and J. Takala, "Indoor localization methods using dead reckoning and 3D map matching," *J. Signal Process. Syst.* **76**(3), 301–312 (2013).
12. J. Biswas and M. Veloso, "Multi-sensor mobile robot localization for diverse environments," *Comput. Sci.* **8371**, 468–479 (2014).
13. J. Blanco, J. González-Jiménez and J. Fernández-Madrigal, "A robust, multi-hypothesis approach to matching occupancy grid maps," *Robotica* **31**(5), 687–701 (2013).
14. S.-H. Kim, C.-W. Jho and H.-K. Hong, "Automatic registration method for multiple 3D range data sets," *J. KISS: Software Appl.* **30**(11,12), 1239–1246 (2003).
15. R., Szymon and M. Levoy, "Efficient Variants of the ICP Algorithm," *Proceedings of the* 3^{rd} *IEEE International Conference on 3-D Digital Imaging and Modeling*, (2001) pp. 145–152.
16. V. H., Sabine and P. Lemmerling, eds. *Total Least Squares and Errors−In−Variables Modeling: Analysis, Algorithms and Applications* (Springer Science & Business Media 2013).
17. P. J. Besl and N. D. Mckay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2) 239–256 (1992).
18. Y. Zhang, J. H.Park and K.T. Chong, "Model algorithm control for path tracking of wheeled mobile robots," *Int. J. Precis. Eng. Manuf.* **11**(5), 705–714 (2010).
19. J. Kim, S. Woo, J. Kim, J. Do, S. Kim and S. Bae, "Inertial navigation system for an automatic guided vehicle with Mecanum wheels," *Int. J. Precis. Eng. Manuf.* **13**(3), 379–386 (2012).
20. C. Hsu, H. Chang and Y. Lu, "Map Building of Unknown Environment using PSO-Tuned Enhanced Iterative Closest Point Algorithm," *Proceedings of the IEEE International Conference on System Science and Engineering (ICSSE)*, (Jul. 2013) pp. 279–284.
21. Y. Lee, J. Song and J. Choi, "Performance improvement of iterative closest point-based outdoor SLAM by rotation invariant descriptors of salient regions," *J. Intell. Robot. Syst.* **71**(3–4), 349–360 (2013).
22. Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.* **13**(2), 119–152 (1994).
23. K. S. Arun, T. S. Huang and S. D. Blostein, "Least square fitting of two 3-d point sets," *IEEE Trans. Pattern Anal. Mach. Intell.* **9**(5), 698–700 (1987).
24. T. Fuiita and Y. Kondo, "3D Terrain Measurement System with Movable Laser Range Finder," *Proceedings of IEEE International Workshop on Safety, Security & Rescue Robotics SSRR*, no. 2, (2009), pp. 1–6.
25. K. Ohno and S. Tadokoro, "Dense 3D Map Building Based on LRF Data and Color Image Fusion," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2005) pp. 2792–2797.
26. F. Martín, R. Triebel, L. Moreno and R. Siegwart, "Two different tools for three-dimensional mapping: DE-based scan matching and feature-based loop detection," *Robotica* **32**(1), 19–41 (2014).
27. H. Surmann, K. Lingemann, A. Nuchter and J. Hertzberg, "A 3D Laser Range for Autonomous Mobile Robots," *Proceedings of the* 32^{nd} *International Symposium on Robotics ISR* (Apr. 2001) pp. 153–158.
28. D. Oritin, J. Neira and J. M. M. Montiel, "Relocation using Laser and Vision," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2 (2004) pp. 1505–1510.
29. P. Núñez, R. Vázquez-Martín, A. Bandera and F. Sandoval, "Fast laser scan matching approach based on adaptive curvature estimation for mobile robots," *Robotica* **27**(3), 469–479 (2009).
30. T. D. Larsen *et al.* "Incorporation of Time Delayed Measurements in a Discrete-Time Kalman Filter," *Proceedings of the* 37^{th} *IEEE Conference on Decision and Control*, vol. 4 (1998) pp. 3972–3977.
31. M. Bak *et al.* "Location Estimation using Delayed Measurements," *Proceedings of the* 5^{th} *IEEE International Workshop on Advanced Motion Control*, Coimbra (1998).

https://doi.org/10.1017/S0263574716000540 Published online by Cambridge University Press