J Intell Robot Syst (2008) 51:461–488
DOI 10.1007/s10846-007-9194-1

UNMANNED SYSTEM PAPER

An Efficient Localization Algorithm Based on Vector Matching for Mobile Robots Using Laser Range Finders

Hee Jin Sohn · Byung Kook Kim

Received: 27 July 2007 / Accepted: 13 November 2007 /
Published online: 7 December 2007

Abstract This paper describes an efficient localization algorithm based on a vector-matching technique for mobile robots with laser range finders. As a reference the method uses a map with line-segment vectors, which can be built from a CAD layout of the indoor environment. The contribution of this work lies in the overall localization process. First, the proposed sequential segmentation method enables efficient vector extraction from scanned data. Second, a reliable and efficient vector-matching technique is proposed. Finally, a cost function suitable for vector-matching is proposed for nonlinear pose estimation with solutions for both nonsingular and singular cases. Simulation results show that the proposed localization method works reliably even in a noisy environment. The algorithm was implemented for our wheelchair-based mobile robot and evaluated in a 135 m long corridor environment.

Keywords Laser range finder · Localization · Line segment extraction · Line segment matching · Map-based navigation

1 Introduction

Autonomous robots are robots that can perform desired tasks in certain environments without human guidance. For a robot to perform the given task autonomously, it must know where it is in order to navigate. Localization is the ability of a robot to estimate its pose (orientation and location) within a given environmental map [28].

Categories (3), (6).

H. J. Sohn (✉) · B. K. Kim
Department of Electrical Engineering and Computer Science,
Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong,
Yuseong-gu, Daejeon, South Korea
e-mail: hjsohn@rtcl.kaist.ac.kr

There are two approaches to robot pose estimation.

- Global localization: The robot pose is determined in global coordinates on a given map without reference to its initial pose. The key to this approach is finding the most probable location on the map that best matches the acquired sensor data. Typical examples include [3, 14, 18].
- Local localization: This approach requires an initial guess of the start pose. If a world map is not provided, the robot updates its pose incrementally while traversing. Because there is no global reference, a small pose error inevitably accumulates, which may lead to a catastrophic pose error after a long traverse. Most scan-matching techniques, such as iterative dual correspondence (IDC) algorithm [19, 27], are in this group. On the other hand, when the robot has been given an environmental map and information about its starting position, it can determine its pose more deterministically. An estimated robot pose is always represented in global coordinates, consistent with the given map, which prevents possible accumulation of pose error.

In this paper, we deal with the map-based local localization problem for mobile robots equipped with laser range finders (LRFs). Various approaches have been proposed and they have the following common key operations.

- *Feature extraction from the sensor data*: Robust and efficient feature extraction is crucial for feature-matching-based localization methods. Various types of features have been used, such as the angle histogram [30], line segment [1, 2, 4, 9–12, 22, 32], and point [29, 31]. Also raw sensor data have been used in many researches without feature extraction [13, 21].
- *Feature matching*: Extracted features are matched to the map for pose estimation. There exist various matching schemes according to the type of features used, but many approaches adopt Mahalanobis distance [2, 6].
- *Pose estimation*: The current robot pose is estimated based on the matching result. For this operation, the least-squares method has been widely used [6, 16, 19, 20].

Line segment is widely used as a main feature for LRF-based localization systems. As LRF scan data are a set of Cartesian points, we require a segmentation method to extract line segments from these. Recently, Nguyen [23] examined six popular segmentation methods and concluded that the split-and-merge and incremental algorithms are preferred because of their superior speed and accuracy. In particular, split-and-merge is better with its lower computational complexity, $O(N \cdot \log N)$ where N is the number of segments. However, we require a more efficient segmentation method for crowded environments where many segments may be extracted, resulting in performance degradation. On the other hand, most matching algorithms for line segment-based localization emphasizes linear characteristics of line segments. For example, a scan and map line segments are determined as a matching pair in [32] when the difference of angle and distance are less than a threshold. Similarly, matching pairs are determined in [6] for line segments showing minimum Mahalanobis distance. However, these approaches do not consider the end points of line segments. It is possible for several map line segments to have almost the same line parameters, which complicates determination of matching pairs with a simple geometric relationship. For the pose estimation process, many successful approaches use nonlinear

optimization methods, e.g., [5, 6, 19, 20], that minimize their own cost functions. The cost function proposed by Beveridge [5] was developed for vision processing, which needs some modification to deal with LRF scan data. In addition, a solution for singularity is not presented in [5]. IDC [19] and MbICP [20] were originally scan-matching algorithms; consequently they focus on minimizing the distance from point to point. For the map-based localization, these scan- matching methods must obtain virtual scan data from the map, requiring additional computation time. WLS [6] uses two types of features: line segments and point features. A line segment is represented as a point on the line that is the closest to the origin. However, as can be seen in Figure 1 of [25], the uncertainty of a line segment increases with distance from the center of rotational uncertainty. When the representative point features exist away from the corresponding line segments, their accuracy cannot be guaranteed.

Our localization algorithm is based on a vector-matching technique: LRF scan data are condensed into a collection of vectors and are matched to an environmental map, which is built from a computer-aided design (CAD) layout of the environment. For efficient vector extraction from raw sensor data, we propose a new sequential least-squares fitting method, which is one of the most efficient methods of line fitting operations. Feature matching is one of the most time-consuming processes in the feature map-based localization. We present an efficient means of reducing the search space. In addition, a new distance measure is proposed for robust vector-matching. Pose estimation is accomplished by the well-known least-squares method. Fast closed-form solutions for both nonsingular and singular cases are derived by considering translation and rotation simultaneously. Simulation and experimental results demonstrate the robustness and accuracy of the proposed localization method compared with other modern techniques.

The remainder of the paper is organized as follows. Section 2 describes the overall localization algorithm based on vector-matching. Section 3 presents simulation and experimental results showing the performance of the proposed algorithm. We present our conclusions in Section 4.

2 Localization Based on Vector-Matching

This section describes the proposed localization algorithm. For reference, the key symbols used in this paper are described in Table 1.

Table 1 List of key symbols used

Symbol	Description
M	A global vector map, which is a set of map vectors
$\mathbf{m}_i$	A map vector defined by a start point q_{i1} and an end point q_{i2}
D	A set of LRF scan points
$\mathbf{s}_i$	A scan vector with a start point r_{i1} and an end point r_{i2}
H	A set of map-scan vector pairs
E	A cost function representing a measure of the distance between matched scan and map vectors
Γ	Lagrange equation for pose estimation

Our localization algorithm is based on the following schemes:

Step 1. Extract vectors from the LRF scan data.
Step 2. Perform vector-matching for map and scan vectors.
Step 3. Estimate the optimum robot pose using the matched vector pairs.

The rest of the Section describes map vectors and the three localization steps.

2.1 Map Vectors

Before explaining details of the three Steps, we explain map vectors, since our localization algorithm adopts a vector map for global pose reference. Vector maps are used implicitly or explicitly by previous researches [2, 7]. The vector map M is a set of map vectors with N_{M} vector elements:

$$\mathrm{M} = \left\{ \mathbf{m}_i | i = 1, \ldots, N_{\mathrm{M}} \right\}. \tag{1}$$

$\mathbf{m}_i$ is a directed line segment – vector – represented by the start point q_{i1} and the end point q_{i2}:

$$\mathbf{m}_i = \left\{ q_{i1}, q_{i2} | q_i \in \mathbb{R}^2 \right\}. \tag{2}$$

It is distinguished from an ordinary line segment as it includes the direction, as shown in Fig. 1. We assume that the left side of a vector is open space and the opposite side is unreachable space.

The main advantage of the vector map is that it simplifies discrimination between the opposite sides of a wall. If a wall is described by two simple line segments $\overline{\mathbf{m}_1}$ and $\overline{\mathbf{m}_2}$ without considering direction, as shown in Fig. 2a, it is ambiguous as to which side of the wall matches the detected scan line segment $\overline{\mathbf{s}_1}$. However, if we use vectors, it is straightforward to determine that the scan vector $\mathbf{s}_1$ matches $\mathbf{m}_1$ by checking directions, as shown in Fig. 2b. We use an LRF that scans the environment counterclockwise; therefore, it is easy to extract vector features from the scan data by using a suitable feature-extraction method.

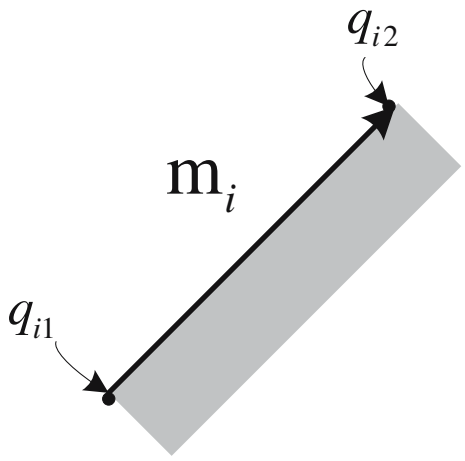

Fig. 1 An example of a map vector. The right side is blocked or filled space and the left side is open space

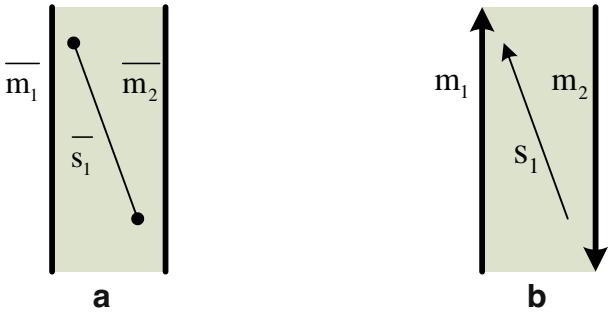

Fig. 2 Matching **a** with line segments, **b** with vectors

2.2 Vector Extraction by the Sequential Segmentation Algorithm

This subsection describes the extraction of vectors from raw scan data (Step 1). Here we propose a new vector extraction algorithm using a sequential least-squares method, which does not require all the points constituting the line to update the line parameters: When a new point is being added, the line parameters are updated incrementally using previously computed information. The complexity of the proposed algorithm is $O(N)$, demonstrating its superior efficiency. The sequential least-squares method is described in Appendix A1.

Given a set of raw scan data $\mathrm{D}=\{d_i | i = 1, \ldots, N_\mathrm{D},\ d_i \in \mathbb{R}^2\}$, we start the segmentation process by finding three consecutive points $\{d_k, \ldots, d_{k+2}\}_{1 \le k \le N_\mathrm{D}-2}$ of which distances $\|d_{k+1} - d_k\|$ and $\|d_{k+2} - d_{k+1}\|$ are within a threshold δ_d and the point variance along the line they constitutes is less than a threshold δ_{σ^2}. Let L_1 be a line which is obtained by least square fitting using the points $\{d_k, \ldots, d_{k+2}\}$:

$$L_1 : x \cdot \cos\theta + y \cdot \sin\theta = \rho. \tag{3}$$

If $\|d_{k+3} - d_{k+2}\|$ is less than δ_d, we calculate the distance $dist(d_{k+3}, L_1)$, which is the perpendicular distance from the scan point $d_{k+3} = (x^d_{k+3}, y^d_{k+3})$ to L_1:

$$dist(d_{k+3}, L_1) = \left| x^d_{k+3} \cdot \cos\theta + y^d_{k+3} \cdot \sin\theta - \rho \right|.$$

If $dist(d_{k+3}, L_1)$ is within the threshold, L_1 is updated including d_{k+3} by the proposed sequential least-squares method, and the next point d_{k+4} is tested. If not, a scan vector $\mathbf{s}_1$ is determined from L_1 and the same process is repeated for $\mathbf{s}_2$ from d_{k+3}. When the i-th line L_i is determined by the scan points $\{d_k, d_{k+1}, \ldots, d_{k+n}\}$, the two terminal points r_{i1}, r_{i2} of $\mathbf{s}_i$ are determined as follows:

$$r_{i1} = (\rho \cdot \cos\theta - t_1 \cdot \sin\theta,\ \ \rho \cdot \sin\theta + t_1 \cdot \cos\theta)$$
$$r_{i2} = (\rho \cdot \cos\theta - t_2 \cdot \sin\theta,\ \ \rho \cdot \sin\theta + t_2 \cdot \cos\theta)$$

where:

$$t_1 = -x^d_k \cdot \sin\theta + y^d_k \cdot \cos\theta$$
$$t_2 = -x^d_{k+n} \cdot \sin\theta + y^d_{k+n} \cdot \cos\theta.$$

The resulting scan vector $\mathbf{s}_i$ is composed of its two end points (r_{i1}, r_{i2}), length l_i, and the standard deviation σ_i of points constituting $\mathbf{s}_i$, as shown in Fig. 3. In addition

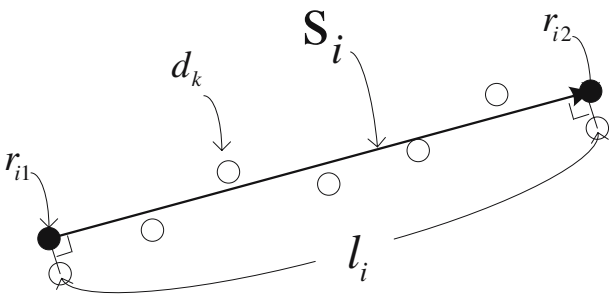

Fig. 3 An example of a scan vector $\mathbf{s}_i$, which has two end points (r_{i1}, r_{i2}) that are obtained during the segmentation process

N_i, the number of scan points contributed for the vector is also included for the pose estimation step:

$$\mathbf{s}_i = \{(r_{i1}, r_{i2}), l_i, \sigma_i, N_i | r_{i1}, r_{i2} \in \mathbb{R}^2, (l_i, \sigma_i) \in \mathbb{R}\}. \tag{4}$$

Note that $\mathbf{s}_i$ is described in the local coordinate system. To be matched to the map vectors, therefore, we have to transform scan, or map, vectors to the general coordinate system. Further descriptions are presented in Section 2.3.

Following MATLAB simulation results show that the proposed sequential segmentation algorithm requires less computation time than the split-and-merge algorithm [24], which is known as one of the most efficient segmentation method [23]. We managed a simulation environment to generate 10 to 50 times of segmentation. Figure 4 shows an example. The robot obtains scan data at the position marked by red arrow (Fig. 4a), which is segmented into 50 vectors (Fig. 4b).

Figure 5 and Table 2 compares the mean computation time for the segmentation algorithms. For each case, 1,000 times of segmentation has been performed. The

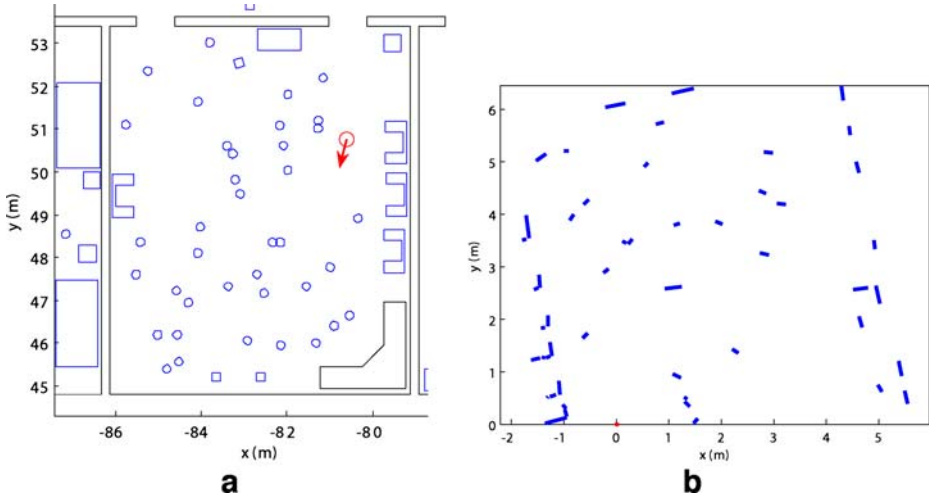

Fig. 4 Simulation for vector extraction. **a** Shows the simulated environment with various obstacles. *Red arrow* indicates the simulated robot. Fifty segments are extracted as shown in (**b**)

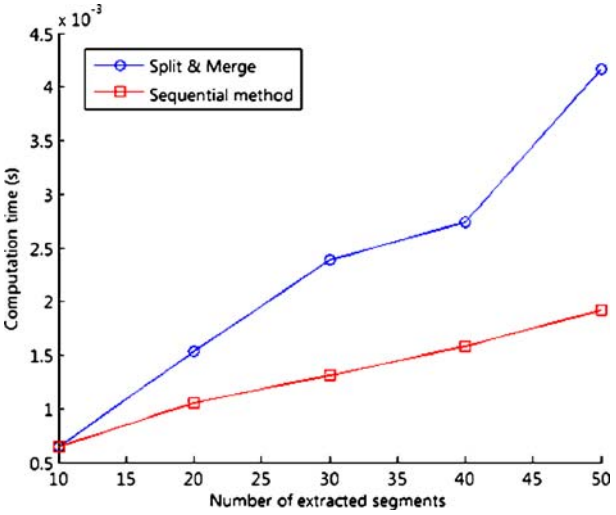

Fig. 5 The computation time of vector extraction by the selected methods. *Blue circle* is for split and merge method and red square is for sequential segmentation method

computation time of split-and-merge increases more rapidly than the proposed method as the number of segmentation increases. From this fact, we can expect that the proposed segmentation algorithm works better in a more cluttered environment.

The uncertainty associated with $\mathbf{s}_i$ is represented by the covariance matrices associated with its end points r_{i1} and r_{i2}. To obtain these parameters, we first calculate the uncertainty of the related line $\mathbf{L}_i$, which is represented by the covariance matrix P_Θ. Here we use Haralick's approach [15] to find P_Θ.

Covariance on $\Theta = [\theta \ \ \rho]^T$, P_Θ, is given by

$$P_\Theta = \left(\frac{\partial g}{\partial \Theta}\right)^{-1} \cdot \left(\Sigma_f + \Sigma_d\right) \cdot \left(\frac{\partial g}{\partial \Theta}\right)^{-1}$$

where:

$$g = \partial E_{fit}/\partial \Theta$$

$$E_{fit} = \sum_{i=1}^{n} f_i^2 = \sum_{i=1}^{n} \left(x_i \cos\theta_n + y_i \sin\theta_n - \rho_n\right)^2$$

Table 2 The computation time of vector extraction by the selected methods

Number of extracted vectors	Split and merge (ms) ...①	Sequential segmentation method (ms) ...②	②/①
10	0.647	0.645	0.997
20	1.533	1.054	0.686
30	2.390	1.314	0.550
40	2.743	1.577	0.575
50	4.161	1.916	0.460

$$\Sigma_f = \sum_{i=1}^{n} \left(\frac{\partial g}{\partial f_i}\right) P_f \left(\frac{\partial g}{\partial f_i}\right)^T$$

$$\Sigma_d = \sum_{i=1}^{n} \left(\frac{\partial g}{\partial d_i}\right) P_{d_i} \left(\frac{\partial g}{\partial d_i}\right)^T$$

To calculate the covariance of end points, $P_{r_{i1},r_{i2}}$, from P_Θ, we adopted Pfister's method [26]. By definition, r_{i1} and r_{i2} are the perpendicular feet of d_k and d_{k+n}, respectively. Therefore, $P_{r_{i1},r_{i2}}$ should be affected by $P_{d_k,d_{k+n}}$ and P_Θ as follows:

$$\begin{aligned} P_{r_{i1}} &= R_{\theta_n} H_a P_s H_a^T R_{\theta_n}^T \\ P_{r_{i2}} &= R_{\theta_n} H_b P_s H_b^T R_{\theta_n}^T \end{aligned} \tag{5}$$

where R_{θ_n} is a rotational matrix, $H_{a,b}$ are selection matrices, and P_s is determined from P_Θ and $P_{d_k,d_{k+n}}$. Further descriptions are presented in Section 4.2.9 of [26].

2.3 Vector-Matching

After scan vectors are extracted, they should be matched to the map vectors to estimate the robot pose (Step 2). There are two important issues in the vector-matching step: efficiency and accuracy. Generally, vector-matching is one of the most time-consuming components in the localization process. Given N_S scan vectors and N_M map vectors, $N_S^{N_M+1}$ possible combination should be tested to find the optimum matching pairs. When a robot operates in a cluttered and complex environment, many scan vectors need to be compared to many map vectors, which requires considerable computation time.

Here we propose a new vector-matching algorithm. A selection guide for matching candidates is presented for the efficiency issue. In addition, a new distance measure is defined that considers the relationship between end points of map vectors and those of scan vectors.

As described in Sections 2.1 and 2.2, scan vectors are defined in the local coordinate frame but the map vectors are described in the global coordinate frame. To find matching pairs, these vectors must be described in the same coordinate system. In general, the number of scan vectors is smaller than the number of map vectors. Therefore, transforming scan vectors to the global coordinate frame costs less than the reverse. For this reason, vector-matching and the following pose estimation processes are performed in the global coordinate system.

Moreover, local localization assumes small or limited odometry error. From this fact, we can limit the candidate map vectors that are to be matched to the scan vectors by considering the sensor range. Figure 6 shows the possible sensing range with respect to each coordinate frames. Throughout this paper, a LMS200 laser range finder is used as the main sensor. It provides 180° coverage with 0.25, 0.5, or 1.0° angular resolution. ψ_{ideal} of Fig. 6a denotes the ideal sensing range in the local coordinate frame.

However, there are some problems remaining. First, the half-circled shape of the sensing range makes selection of candidate map vectors difficult. For efficiency, it is simplified to be the rectangle ψ_L, which contains ψ_{ideal}, considering the odometry error $\{\delta_x^\psi, \delta_y^\psi, \delta_\phi^\psi\}$. Then ψ_L is transformed into global coordinates (ψ_L' of Fig. 6b). To determine candidate map vectors, all map vectors must be tested whether they

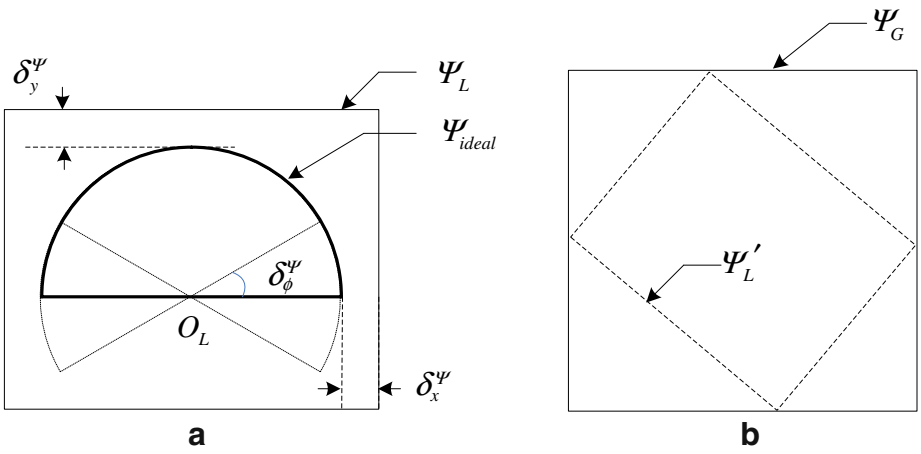

Fig. 6 Possible sensing area with respect to each coordinate frames. **a** ψ_{ideal} is the ideal sensing area in the local coordinate frame, which is simplified as ψ_L for efficiency. Here O_L is the local origin and $\{\delta_x^{\psi}, \delta_y^{\psi}, \delta_{\phi}^{\psi}\}$ indicates odometry error. **b** ψ_L is transformed into global coordinate frame becoming ψ_L', which is further simplified as ψ_G for algorithmic efficiency

are partially or totally included in ψ_L'. The computational complexity is greatly dependent on the shape of ψ_L'. Further simplification of ψ_L' into ψ_G makes this operation extremely simple.

Figure 7 and Table 3 shows the simulation results for the vector-matching operation using the full search method and the proposed algorithm. It can be seen that the proposed algorithm becomes more efficient as the number of map vectors increases.

As mentioned in Section 1, considering only line parameters (angle and perpendicular distance from the origin) is not enough for vector-matching. Figure 8 shows an example of matching ambiguity. Here $\mathbf{m}_1$ and $\mathbf{m}_5$ lies on the same line: If we consider only line parameters, we can not determine which of them is closer to $\mathbf{s}_1$. On the other

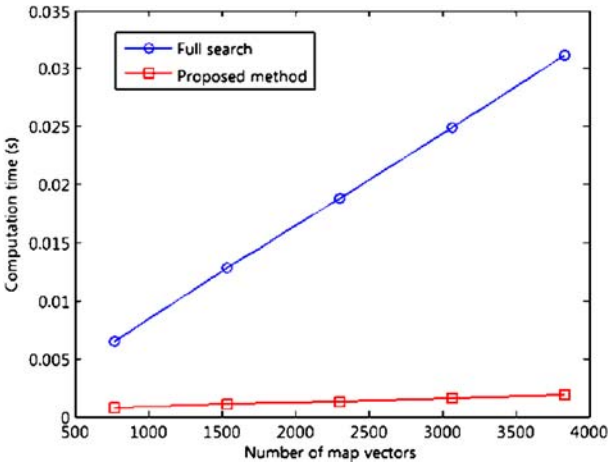

Fig. 7 Average computation time of the selected vector-matching methods with respect to the number of map vectors. *Blue circle* is for the full search method and red square is for the proposed method

Table 3 Average computation time of the selected vector-matching methods with respect to the number of map vectors

N_M	Full search (ms) ...①	Proposed method (ms) ...②	②/①
767	6.536	0.767	0.174
1534	12.814	1.041	0.081
2301	18.764	1.315	0.070
3068	24.943	1.619	0.065
3835	31.198	1.938	0.062

hand, $\mathbf{m}_3$ is farther from $\mathbf{s}_1$ than other vectors considering line parameters. However, it needs less amount of translation and rotation to be matched with $\mathbf{s}_1$ compared to $\mathbf{m}_1$ or $\mathbf{m}_5$.

To solve this kind of ambiguity problem, we present a new distance measure considering geometric relationship between vectors. Let M′ be a set of map vectors that are partially or totally included in ψ_G.

$$\mathrm{M}' = \{\mathrm{m}_j | j = 1, \ldots, N_{\mathrm{M}'}\} \tag{6}$$

Then a map vector $\mathbf{m}_j \in \mathrm{M}'$ is determined to be a match for a scan vector $\mathbf{s}_i$ when it minimizes the following distance measure:

$$dist^2(\mathbf{s}_i, \mathbf{m}_j) \triangleq dist^2(r_{i1}, \mathbf{m}_j) + dist^2(r_{i2}, \mathbf{m}_j), \tag{7}$$

where $r_{ik|_{k=1,2}}$ are the end points of $\mathbf{s}_i$.

Let the projection of r_{ik} onto $\mathbf{m}_j$ be r_{ik}^p as described in Fig. 9. If r_{ik}^p is on the outside of $\mathbf{m}_j$ (Fig. 9a, c), $dist(r_{ik}, \mathbf{m}_j)$ is the shorter one between $\|r_{ik} - q_{j1}\|$ and $\|r_{ik} - q_{j2}\|$. If r_{ik}^p exists on $\mathbf{m}_j$, $dist(r_{ik}, \mathbf{m}_j)$ becomes $\|r_{ik} - r_{ik}^p\|$. That is, $dist^2(\mathbf{s}_i, \mathbf{m}_j)$ explains not only vertical but also horizontal distance between vectors, which clarify the ambiguity problem of Fig. 8. These facts can be expressed in a simple formulation:

$$\begin{aligned} dist^2(r_{ik|_{k=1,2}}, \mathbf{m}_j) &\triangleq \min_{\lambda \in [0,1]} dist^2(r_{ik|_{k=1,2}}, (1-\lambda)q_{j1} + \lambda q_{j2}) \\ &= \min_{\lambda \in [0,1]} (a\lambda^2 + b\lambda + c), \end{aligned} \tag{8}$$

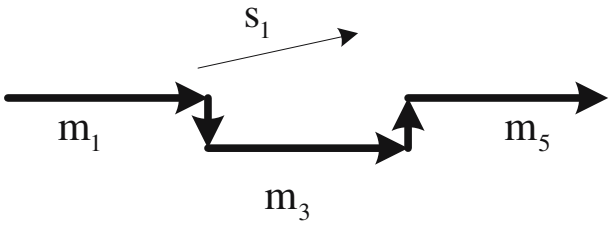

Fig. 8 An example of matching ambiguity

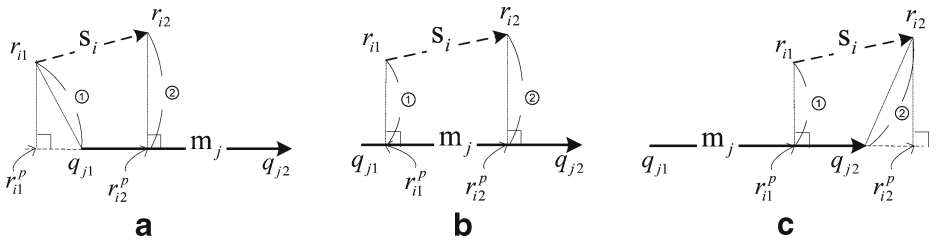

Fig. 9 Distance measures for vector-matching according to the geometry of vectors. ① indicates $dist(r_{i1}, \mathbf{m}_j)$ and ② is for $dist(r_{i2}, \mathbf{m}_j)$. $\mathbf{s}_i$ is a scan vector with end points (r_{i1}, r_{i2}), and $\mathbf{m}_j$ is a map vector with two points (q_{j1}, q_{j2}). r_{i1}^p and r_{i2}^p are perpendicular feet of r_{i1} and r_{i2} onto $\mathbf{m}_j$ respectively

where:

$$a = \|q_{j2} - q_{j1}\|^2$$
$$b = -2(r_{ik}|_{k=1,2} - q_{j1})^T (q_{j2} - q_{j1})$$
$$c = \|r_{ik}|_{k=1,2} - q_{j1}\|^2.$$

Table 4 shows the complete vector-matching process. As a result, we obtain a set of N matching pairs H defined as:

$$\mathrm{H} = \{h_k | h_k =< \mathrm{s}_i, \mathrm{m}_j >,\ \mathrm{s}_i \in S,\ \mathrm{m}_j \in \mathrm{M}',\ k = 1, \ldots, N\}.$$

2.4 Nonlinear Pose Estimation

The purpose of the pose estimation step is to find the most plausible robot pose by calculating the rotation ϕ and translation T that optimally matches the scan data to the map (Step 3). In this subsection, a cost function for the proposed vector-matching is presented, and a closed-form solution is also presented.

Table 4 The proposed vector-matching algorithm

Pseudo-code for the vector-matching algorithm

```
Determine M'
For all scan vectors {s_i}_{i=1,...,N_S}
    For all map vectors {m_j}_{j=1,...,N_M'}
        If |∠s_i − ∠m_j| is under threshold
            Compute dist^2(s_i, m_j)
            If dist^2(s_i, m_j) is under threshold and is minimum
                m_min = m_j
            End If
        End If
    End For
    If m_min is nonzero
        H ← h_k =< s_i, m_j >
    End If
End For
```

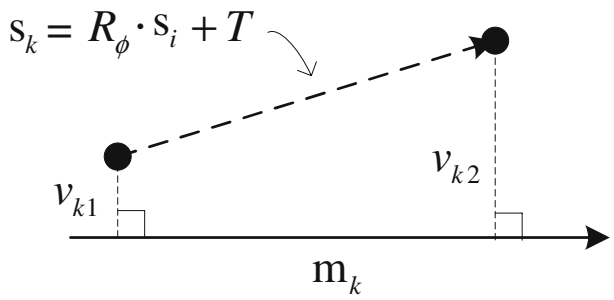

Fig. 10 Parameters for pose calculation. $v_{kj}|_{j=1,2}$ are perpendicular distances from the end points of $\mathbf{s}_k$ to $\mathbf{m}_k$

From the result of the previous matching step, we can build a cost function E as:

$$E = \eta \cdot \sum_{k=1}^{N} \frac{N_k l_k}{\sigma_k} \sum_{j=1}^{2} v_{kj}^2 \tag{9}$$

where N_k, l_k and σ_k are entities of $\mathbf{s}_k$ as in Eq. 4. These parameters indicate the quality of the scan vector for localization. A vector receives greater weight when it is either made from more scan points or longer than other vectors. In addition, point variance is also considered to bestow a penalty on noisy vectors [17]. η is a normalizing constant.

Here, v_{kj} is defined as a perpendicular distance from $r_{k(1,2)}$ of $\mathbf{s}_k$, which is transformed from $\mathbf{s}_i$ by rotation ϕ and translation T, to the corresponding map vector $\mathbf{m}_k$ (Fig. 10):

$$v_{kj} = \begin{bmatrix} \cos\theta'_k & \sin\theta'_k \end{bmatrix} \cdot \left(R_\phi \cdot r'_{kj} + T \right) - \rho'_k \tag{10}$$

where θ_k and ρ_k are line parameters for the map vector $\mathbf{m}_k$.

Using the following substitution:

$$\Theta_k = \begin{bmatrix} \cos\theta'_k \\ \sin\theta'_k \end{bmatrix}, \quad \Delta_{kj} = \begin{bmatrix} x'_{kj} & -y'_{kj} \\ y'_{kj} & x'_{kj} \end{bmatrix}, \quad \Phi = \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix}$$

v_{kj} can be simplified as:

$$v_{kj} = \Theta_k^T \cdot (\Delta_{kj}\Phi + T) - \rho'_k. \tag{11}$$

Expanding Eq. 9 with Eq. 11 gives:

$$\begin{aligned} E(\Phi, T) &= \eta \cdot \sum_{k=1}^{N} \frac{N_k l_k}{\sigma_k} \sum_{j=1}^{2} \left(\Theta_k^T \cdot (\Delta_{kj}\Phi + T) - \rho'_k \right)^2 \\ &= \Phi^T A \Phi + 2T^T B \Phi + T^T C T - 2U^T \Phi - 2V^T T + \tau, \end{aligned} \tag{12}$$

where:

$$A = \eta \cdot \sum_{k=1}^{N} \frac{N_k l_k}{\sigma_k} \sum_{j=1}^{2} \Delta_{kj}^T \Theta_k \Theta_k^T \Delta_{kj} \qquad B = \eta \cdot \sum_{k=1}^{N} \frac{N_k l_k}{\sigma_k} \sum_{j=1}^{2} \Theta_k \Theta_k^T \Delta_{kj},$$

$$C = 2\eta \cdot \sum_{k=1}^{N} \frac{N_k l_k}{\sigma_k} \Theta_k \Theta_k^T, \qquad U = \eta \cdot \sum_{k=1}^{N} \frac{N_k l_k}{\sigma_k} \sum_{j=1}^{2} \rho_k' \Delta_{kj}^T \Theta_k,$$

$$V = 2\eta \cdot \sum_{k=1}^{N} \frac{N_k l_k}{\sigma_k} \rho_k' \Theta_k, \qquad \tau = 2\eta \cdot \sum_{k=1}^{N} \frac{N_k l_k}{\sigma_k} \left(\rho_k'\right)^2. \tag{13}$$

Because Eq. 12 is a quadratic form, the local minima of $E(\Phi, T)$ satisfy:

$$\frac{\partial E}{\partial \Phi} = 0 \tag{14}$$

$$\frac{\partial E}{\partial T} = 0. \tag{15}$$

Φ is constrained by the following trigonometric identity:

$$\Phi^{\mathrm{T}} \Phi = 1, \tag{16}$$

Now Eq. 15 is combined with Eq. 12 to reveal:

$$\frac{\partial E}{\partial T} = 2B\Phi + 2CT - 2V = 0. \tag{17}$$

Now the solutions for Eq. 17 are categorized by the singularity of matrix C. If C is a nonsingular matrix, the unique translation T is given as a function of Φ:

$$T = C^{-1}(V - B\Phi). \tag{18}$$

Infinitely many solutions for T exist satisfying the constraint (17) if C is singular. Figure 11 shows this situation. The robot pose is uniquely determined if there exist at least two scan segments that are not parallel to each other, as in Figs. 11a and 11b. However, when only parallel scan segments are detected, as in Figs. 11c and 11d, infinitely many solutions exist along a line. The singularity of C in each case is described in Appendix A2. The following subsections describe these situations.

2.4.1 When C is a Nonsingular Matrix

When the scan vectors are not parallel to each other, the cost function E has a unique minimum [17]. Substituting T of Eq. 12 for Eq. 18 gives:

$$E = \Phi^{\mathrm{T}} P \Phi + 2W^{\mathrm{T}} \Phi + \tau', \tag{19}$$

where:

$$P = A - B^{\mathrm{T}} (C^{-1})^{\mathrm{T}} B$$

$$W = B^{\mathrm{T}} (C^{-1})^{\mathrm{T}} V - U$$

$$\tau' = \tau - V^{\mathrm{T}} (C^{-1})^{\mathrm{T}} V.$$

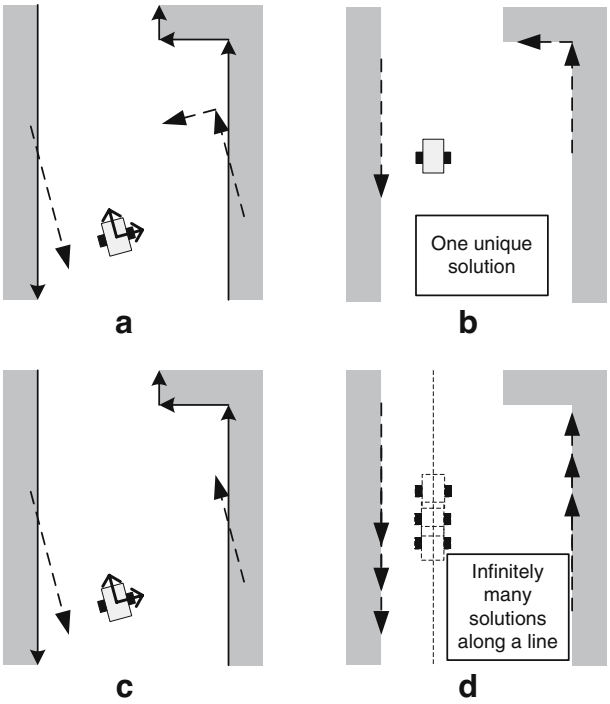

Fig. 11 Situations for a unique solution and infinite solutions. *Dashed vectors* denote scan vectors obtained at the robot pose shown. When the scan vectors are not parallel to each other, as in (**a**), there exists a unique solution (**b**). However, when only parallel scan vectors are obtained, as in (**c**), infinitely many solutions exist (**d**)

Now, Φ is the only variable in cost E. To find the optimum ϕ minimizing E, we form a Lagrangian combining Eqs. 19 and 16, for example:

$$\Gamma = E(\Phi, T) + \lambda(\Phi^T \Phi - 1). \tag{20}$$

The value of Φ minimizing Γ satisfies:

$$\frac{\partial \Gamma}{\partial \Phi} = 2P\Phi + 2W + 2\lambda\Phi = 0$$

or:

$$(P + \lambda I)\Phi + W = 0. \tag{21}$$

Substituting P, W, and Φ of Eq. 21 with the following definitions:

$$P = \begin{bmatrix} p_1 & p_3 \\ p_3 & p_2 \end{bmatrix}, \; W = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

and excluding λ from Eq. 21 yields:

$$p_3 \left(\cos^2 \phi - \sin^2 \phi\right) + (p_2 - p_1) \cos\phi \sin\phi + w_2 \cos\phi - w_1 \sin\phi = 0. \tag{22}$$

Further expansion of Eq. 22, using identity Eq. 16, leads to a quartic in $\sin\phi$ as:

$$k_4 \sin^4\phi + k_3 \sin^3\phi + k_2 \sin^2\phi + k_1 \sin^1\phi + k_0 = 0, \tag{23}$$

where:

$$\begin{aligned} k_4 &= (p_1 - p_2)^2 + 4p_3^2 \\ k_3 &= -2(p_1 - p_2)w_2 + 4p_3 w_1 \\ k_2 &= w_1^2 + w_2^2 - k_4 \\ k_1 &= 2(p_1 - p_2)w_2 - 2p_3 w_1 \\ k_0 &= p_3^2 - w_2^2. \end{aligned}$$

As described above, ϕ is the angular difference between the odometry and the real pose. Assuming moderate odometry error, ϕ should be $|\phi| < \pi/2$ where $\sin\phi$ is a one-to-one function. Among all the real roots of Eq. 23, we can always find a unique solution that is under a threshold value and satisfies Eq. 22. As a result, the translation T is also determined uniquely from Eq. 18.

2.4.2 When C is a Singular Matrix

If the scan vectors obtained are parallel to each other, or only one vector is detected, as shown in Fig. 11c, Eq. 12 will produce a unique rotation ϕ and infinitely many candidates for translation T. When only parallel vectors are obtained, matrices B and C in Eq. 13 become singular, as shown in Appendix A2. We define the individual elements of all matrix parameters of Eq. 17 as follows:

$$C = \begin{bmatrix} c_1 & c_3 \\ c_3 & c_2 \end{bmatrix}, B = \begin{bmatrix} b_1 & b_2 \\ \frac{c_3}{c_1} b_1 & \frac{c_3}{c_1} b_2 \end{bmatrix}, T = \begin{bmatrix} t_2 \\ t_2 \end{bmatrix}, V = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \tag{24}$$

Expanding Eq. 17 with Eq. 24 results in a line equation for a possible translation T:

$$c_1 t_1 + c_3 t_2 = v_1 - b_1 \cos\phi - b_2 \sin\phi. \tag{25}$$

We build a Lagrangian from Eq. 12, as in Eq. 20. Substitution of:

$$A = \begin{bmatrix} a_1 & a_3 \\ a_3 & a_2 \end{bmatrix}, \quad U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

and using a similar expansion to that in Eq. 22 gives:

$$k_4 \sin^4\phi + k_3 \sin^3\phi + k_2 \sin^2\phi + k_1 \sin^1\phi + k_0 = 0, \tag{26}$$

where:

$$k_4 = \left(a_2 - a_1 + \frac{b_1^2 - b_2^2}{c_1}\right)^2 + 4\left(a_3 - \frac{b_1 b_2}{c_1}\right)^2$$

$$k_3 = 4\left(a_3 - \frac{b_1 b_2}{c_1}\right)\left(\frac{b_1}{c_1}v_1 - u_1\right) + 2\left(a_2 - a_1 + \frac{b_1^2 - b_2^2}{c_1}\right)\left(\frac{b_2}{c_1}v_1 - u_2\right)$$

$$k_2 = -\left(a_2 - a_1 + \frac{b_1^2 - b_2^2}{c_1}\right)^2 - 4\left(a_3 - \frac{b_1 b_2}{c_1}\right)^2 + \left(\frac{b_1}{c_1}v_1 - u_1\right)^2 + \left(\frac{b_2}{c_1}v_1 - u_2\right)^2$$

$$k_1 = -2\left(a_3 - \frac{b_1 b_2}{c_1}\right)\left(\frac{b_1}{c_1}v_1 - u_1\right) - 2\left(a_2 - a_1 + \frac{b_1^2 - b_2^2}{c_1}\right)\left(\frac{b_2}{c_1}v_1 - u_2\right)$$

$$k_0 = \left(a_3 - \frac{b_1 b_2}{c_1}\right)^2 - \left(\frac{b_2}{c_1}v_1 - u_2\right)^2.$$

Now, the unique rotation ϕ is calculated, and the line equation for translation is also obtained, by inserting ϕ into Eq. 25. The resultant line equation for possible translations T is:

$$\begin{aligned} c_1 t_1 + c_3 t_2 &= v_1 - b_1 \cos\phi - b_2 \sin\phi \\ \Rightarrow \cos\psi \cdot t_1 + \sin\psi \cdot t_2 &= \epsilon, \end{aligned} \tag{27}$$

where $\cos\psi = \frac{c_1}{\sqrt{c_1^2+c_3^2}}$, $\sin\psi = \frac{c_3}{\sqrt{c_1^2+c_3^2}}$, and $\epsilon = \frac{v_1 - b_1 x - b_2 y}{\sqrt{c_1^2+c_3^2}}$.

Let the optimal translation T be the perpendicular foot of the presumed robot position, known by odometry, onto the line of Eq. 27, which means that T is the perpendicular foot of the origin:

$$T = \begin{bmatrix} \epsilon \cos\psi \\ \epsilon \sin\psi \end{bmatrix}. \tag{28}$$

Note that the resultant robot pose can be calculated by transforming ϕ and T into the global coordinate system.

2.4.3 Uncertainty

The uncertainty of the robot pose is obtained in a similar manner, as shown in Section 2.2:

$$P_{x,y,\phi} = \left(\frac{\partial h}{\partial(x, y, \phi)}\right)^{-1} \cdot (\Sigma_v + \Sigma_r + \Sigma_m) \cdot \left(\frac{\partial h}{\partial x, y, \phi}\right)^{-1} \tag{29}$$

where $h = \partial E/\partial(x, y, \phi)$ and

$$\Sigma_v = \sum_{k=1}^{N} \sum_{j=1}^{2} \frac{\partial h}{\partial v_{kj}} \cdot P_v \cdot \frac{\partial h}{\partial v_{kj}}^T$$

$$\Sigma_r = \sum_{k=1}^{N} \sum_{j=1}^{2} \frac{\partial h}{\partial r_{kj}} \cdot P_{r_{kj}} \cdot \frac{\partial h}{\partial r_{kj}}^T$$

$$\Sigma_m = \sum_{k=1}^{N} \frac{\partial h}{\partial m_k} \cdot P_{m_k} \cdot \frac{\partial h}{\partial m_k}^T$$

Here P_{m_k} is given as an input, and P_v is estimated as the sample covariance, which is the same as $E(x, y, \phi)$. $P_{r_{kj}}$ is the variance of two end points of s_k, which is described in Section 2.2.

3 Simulations and Experiments

In this section, the performance of the proposed localization method is evaluated by simulations and experiments.

3.1 Simulations

Two famous localization methods were selected for comparison with the proposed localization algorithm. Borges [8] presented the WLS estimator. This method uses the cost function $J(\tilde{z}, U)$, which is similar to Eq. 9. However, the main feature used in Borges's method is different from ours; he used infinite line and point features for pose estimation. As shown in [26], the uncertainty of a scan vector is a minimum at the center of mass and maximum at the two terminal points. For this reason, the features used in [8] may suffer more uncertainty compared to the proposed vector

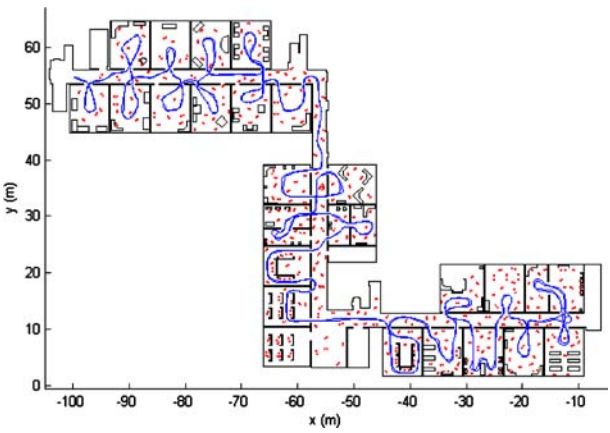

Fig. 12 The clean environment of which obstacle ratio is 15.67%. Obstacles are represented as *red circles and rectangles*. The simulated robot is initiated at a random position on the *blue curve* and starts global pose estimation navigating *along the trajectory*

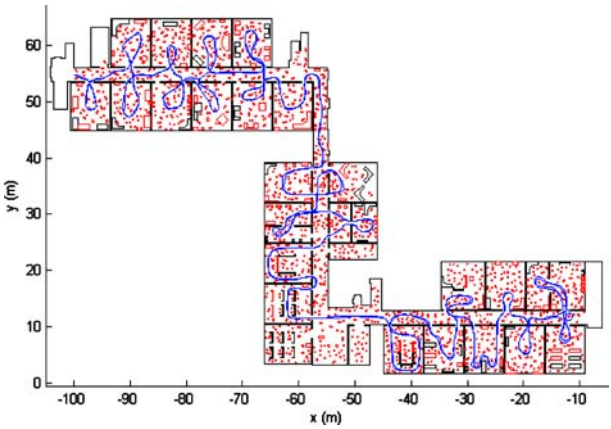

Fig. 13 The heavily occupied environment. The ratio of corrupted sensor data increases to an average of 69.11%

feature. In addition, IDC [19] was compared to our method to determine whether the scan- matching method is useful for map-based localization.

We managed two simulation environments with different obstacle ratios. Here we use the term 'obstacle ratio' as the ratio of scan data that is obtained from objects that are not present in the map. The clean environment of Fig. 12 contains few obstacles with a ratio of 15.67%. On the other hand, Fig. 13 shows a heavily occupied environment, for which the obstacle ratio is 69.11%. The more obstacles in the environment, the greater the chance of matching error.

It was assumed that the robot moves along the predefined trajectory with a velocity of 0.3 m/s. Odometry is modeled that suffers from random noise by 10% of real wheel rotation. The range sensor was modeled on the SICK LMS-200, which is the same as fitted to our mobile robot.

Table 5 shows the result of the pose estimation simulation. For clarity, only position error is presented. As can be seen, the proposed method shows minimum mean error compared to the other algorithms, which shows the adequacy of vector feature used and the accuracy of the pose estimation.

Figure 14 shows $\pm 3\sigma$ confidence intervals for the proposed localization method. As can be seen, the proposed approach presents consistent uncertainty estimates with pose error.

3.2 Experiments

The proposed localization technique was implemented on a two-wheeled autonomous wheelchair, as shown in Fig. 15. Our system is based on a commercial

Table 5 A simulation result for the performance of selected pose estimation algorithms

	Clean		Heavily occupied	
	Mean err(m)	Std. dev.	Mean err(m)	Std. dev.
Proposed	0.0092	0.0363	0.0208	0.0671
WLS	0.0347	0.0480	0.0479	0.0602
IDC	0.0313	0.0736	0.0412	0.0806

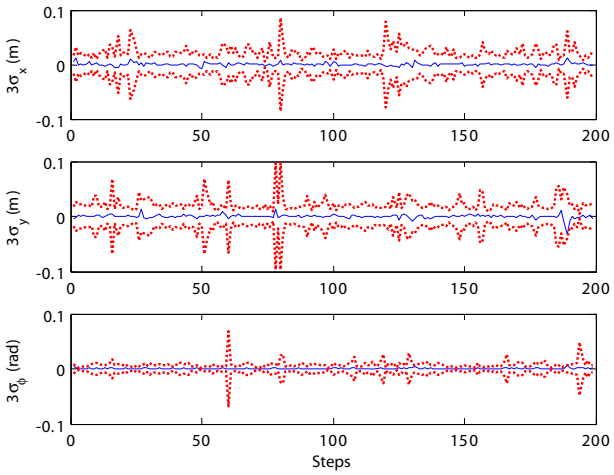

Fig. 14 Errors (*blue line*) and $\pm 3\sigma$ confidence intervals (*red dots*) given by the proposed localization approach in the heavily occupied environment

battery-powered wheelchair *Chairman*, manufactured by Permobil (see http://www.permobil.com), which has two drive motors, four seat actuator motors, and three control modules: a joystick module that receives the user's instructions and converts it to a corresponding control command, a power module that converts the

Fig. 15 The wheelchair mobile robot

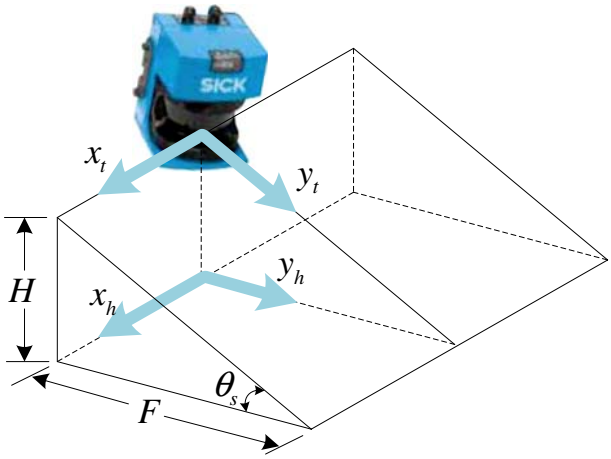

Fig. 16 The geometry of the measurement system. The SICK LMS-200 on the chassis is tilted down by $\theta_s = 25°$ at a height of $H = 169$ cm

control command to power signals to drive the wheelchair and transmits the current status to the joystick module, and a seat-actuator module that provides the power for electrical adjustment of the seat position. These modules communicate via RS232c serial interfaces.

Our retrofitted robotic wheelchair is equipped with an industrial PC (CPU: Pentium III-1 GHz, RAM: 512 MB), mounted on the back of the wheelchair, and running Linux 2.4.22 with RTAI 24.1.10 for real-time processing. An LRF (SICK LMS 200) is mounted on a custom-designed aluminum prop, at a height of 169 cm, and with its scan plane tilted at an angle of 25° to the ground, so that the wheelchair can perceive the environment and detect objects below the height of the LRF (Fig. 16). The x-y coordinates of scanned data are corrected by:

$$x_h = x_t$$

$$y_h = y_t \cdot \cos\theta_s$$

where (x_t, y_t) is the original point and (x_h, y_h) is the corrected point. Note that the scan points for which y_h is approximately F are eliminated to prevent erroneous wall detection.

The LRF provides a selectable angular resolution of 0.25°, 0.5°, or 1°. In our system, it scans through 180°, at 0.5° angular resolution, with each complete scan taking 26 ms. Because the data packet size for one complete scan is 733 bytes, it is linked to the PC via a RS422 serial port for high-speed data transmission at 500 Kbps.

Figure 17 shows the block diagram of the implemented localization system; a shaded block denotes physical hardware. Raw scan data obtained from the LRF are corrected according to the geometry of the LRF in the 'Data correction' block. The 'Vector extraction' block performs sequential least-squares fitting for vector extraction. Then scan vectors are transformed to the global coordinates according to the estimated pose by dead reckoning and compared to the map vectors to find matching vector pairs. Finally, the 'Pose estimation' block performs nonlinear optimization to determine the resultant robot pose.

While traversing a 135 m corridor on the third floor of Dept. of EE in KAIST, the wheelchair system scanned the environment 14,838 times. The corridor has two

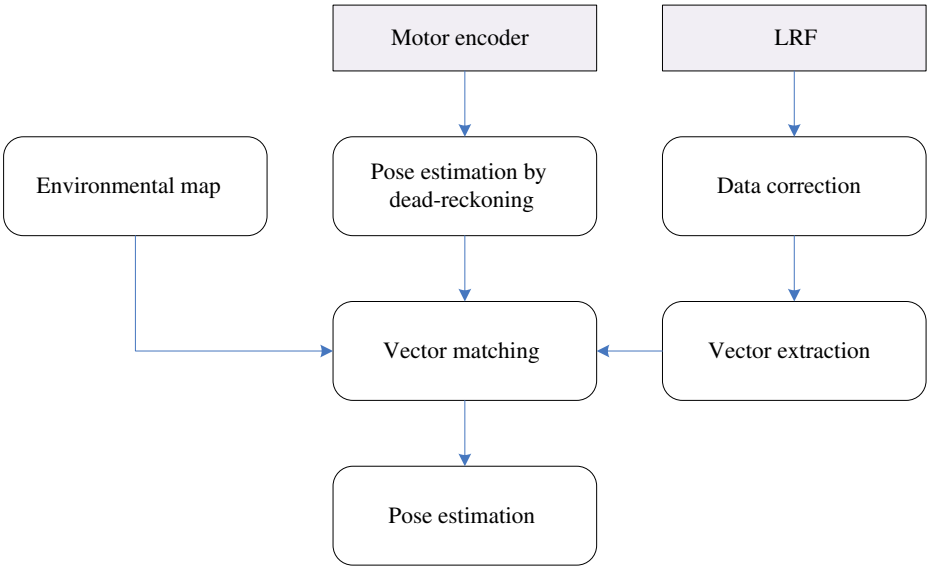

Fig. 17 A block diagram of the implemented localization system

corners and 41 doors, some of which were open and others closed. Some sections have glass windows on the wall, where the laser beam may penetrate or may be reflected in an undefined manner. At each scan time, the robot always met a new environment, i.e., it did not pass a previously passed location.

A map of the environment was built from the CAD layout as a set of vectors, which does not include any sophisticated information. All doors were assumed to be closed and glass windows were treated as ordinary walls, meaning that the robot cannot travel through them. In fact, some doors were open and some glass windows were not detected by the LRF. Moreover, there were obstacles not described in the map, such as chairs, flower pots, bookshelves, etc., and some parts of the corridor were slightly different from the CAD layout, adding considerable uncertainty to the localization process.

Our localization algorithm proved to have real-time performance. As Table 6 shows, the computation time for localization is less than 1 ms, which is less than 1/10 of the data transmission time. In addition, the total processing time, including data transmission time, is 17.33 ms, which allows the whole localization process to be completed within the LRF scan time of 26 ms.

Table 6 Average computation time for each localization step

	Average computation time (ms)
Data transmission	16.1
Vector extraction	0.2
Vector-matching	0.9
Pose calculation	0.1
Total	17.3

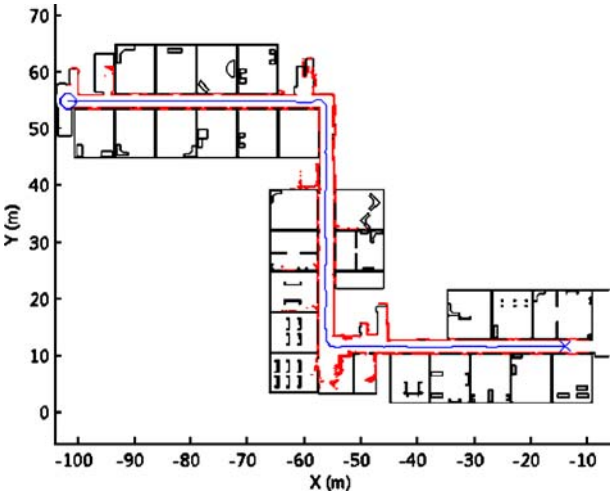

Fig. 18 Overlaid scan data according to the localization results given by the proposed method (red dots). The mobile robot starts navigation at the position 'O' and arrives at 'X'

Figure 18 shows the localization result given by the proposed approach. It is not possible to know the real robot pose at every localization step. To examine the algorithmic performance, all scan data are overlaid on the map; the data were transformed to the global coordinate system according to the corresponding estimated pose. Related uncertainty is shown in Fig. 19. Here the $\pm 3\sigma$ bound for the y-axis is larger than that of the x-axis because most parts of the environment are corridors with few doors open, which results in many singular cases. However, the uncertainty still remains under the $\pm 0.5m$ boundary.

An experiment for room-to-room localization has been performed as shown in Fig. 20. The robot starts moving at $(0, 0)$ in Room A, navigates to Room B and comes back. It moved 65.31 m and performed 3,813 steps of localization. In this

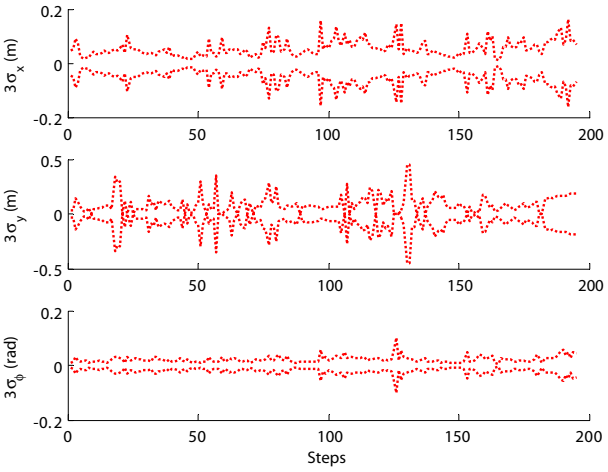

Fig. 19 Confidence intervals ($\pm 3\sigma$) given by the proposed localization approach in a real environment

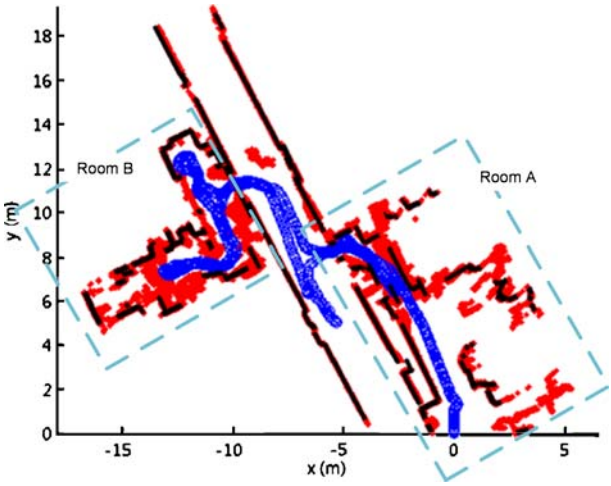

Fig. 20 Experimental result of room-to-room localization. Map (*black line segment*) is built by the proposed vector-matching method which is overlaid with corresponding scan data (*red dots*). The robot path is depicted as *blue circles*

case, environmental map is built by the proposed vector-matching algorithm, which suggests that efficient SLAM should be possible with the aid of our localization method.

4 Conclusions

This paper presents an efficient, fast, accurate, and robust localization algorithm, based on a vector-matching technique, for mobile robots equipped with LRFs. A vector is composed of two end points indicating its direction, which ensure robust and accurate matching results in our localization process. The proposed localization technique is performed in three steps: First, a sequential segmentation method is introduced, which enables fast and reliable segmentation of the scan data. Second, a new vector-matching method is presented that performs robustly and efficiently even with a map composed of a large number of vectors. Finally, a cost function is proposed with closed-form solutions for the optimality equation, which ensures the best result with vector features. Simulation and experimental results have verified that the proposed localization method works accurately and robustly, given the occurrence of unexpected obstacles not present on the map.

Future work includes the expansion of the algorithm to the global localization problem, because there will be many situations where the robot cannot be provided with information about its initial position. In addition, efficient SLAM is possible with the aid of the localization method presented here.

Appendix

A1 Sequential Least-Squares Fit

Lu [19] described a line-fitting algorithm using the least-squares method. Here we extended it to be able to operate in a sequential manner.

A collection of points $\{(x_i, y_i)|i = 1, \ldots, n\}$ is to be fitted to a line L_n: $x \cdot \cos\theta_n + y \cdot \sin\theta_n = \rho_n$ in a least-squares sense. (θ, ρ) minimizes the following error:

$$E_{fit} = \sum_{i=1}^{n} (x_i \cos\theta_n + y_i \sin\theta_n - \rho_n)^2. \tag{30}$$

A closed form solution is given by:

$$\begin{aligned} \theta_n &= \frac{1}{2}\text{atan2}\frac{-2\left(S_n^{xy} - n\overline{x}_n\overline{y}_n\right)}{\left(S_n^{yy} - S_n^{xx}\right) - n\left(\overline{y}_n^2 - \overline{x}_n^2\right)} \\ \rho_n &= \overline{x}_n \cos\theta_n + \overline{y}_n \sin\theta_n \end{aligned} \tag{31}$$

where:

$$\begin{aligned} S_n^{xx} &= \sum_{i=1}^{n} x_i^2, \quad S_n^{yy} = \sum_{i=1}^{n} y_i^2, \quad S_n^{xy} = \sum_{i=1}^{n} x_i y_i, \\ \overline{x}_n &= \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \overline{y}_n = \frac{1}{n}\sum_{i=1}^{n} y_i. \end{aligned}$$

When a point (x_{n+1}, y_{n+1}) is added, the updated line equation becomes L_{n+1}: $x \cdot \cos\theta_{n+1} + y \cdot \sin\theta n + 1 = \rho n + 1$, where θ_{n+1} and ρ_{n+1} are given by:

$$\begin{aligned} \theta_{n+1} &= \frac{1}{2}\text{atan2}\frac{-2\left(S_{n+1}^{xy} - (n+1)\overline{x}_{n+1}\overline{y}_{n+1}\right)}{\left(S_{n+1}^{yy} - S_{n+1}^{xx}\right) - (n+1)\left(\overline{y}_{n+1}^2 - \overline{x}_{n+1}^2\right)} \\ \rho_{n+1} &= \overline{x}_{n+1} \cos\theta_{n+1} + \overline{y}_{n+1} \sin\theta_{n+1} \end{aligned} \tag{32}$$

Using the substitution:

$$\begin{aligned} S_{n+1}^{xx} &= S_n^{xx} + x_{n+1}^2, \quad S_{n+1}^{yy} = S_n^{yy} + y_{n+1}^2, \quad S_{n+1}^{xy} = S_n^{xy} + x_{n+1}y_{n+1}, \\ \overline{x_{n+1}} &= \frac{n\overline{x_n} + x_{n+1}}{n+1}, \quad \overline{y_{n+1}} = \frac{n\overline{y_n} + y_{n+1}}{n+1}, \end{aligned}$$

we obtain a sequential least-squares fit equation:

$$\begin{aligned} \theta_{n+1} &= \frac{1}{2}\text{atan2}\frac{-2\left(S_n^{xy} + x_{n+1}y_{n+1} - \frac{1}{n+1}\left(n\overline{x}_n + x_{n+1}\right)\left(n\overline{y}_n + y_{n+1}\right)\right)}{\left(S_n^{yy} - S_n^{xx}\right) + \left(y_{n+1}^2 - x_{n+1}^2\right) - \frac{1}{n+1}((n\overline{y}_n + y_{n+1})^2 - (n\overline{x}_n + x_{n+1})^2)} \\ \rho_{n+1} &= \frac{n\overline{x}_n + x_{n+1}}{n+1}\cos\theta_n + \frac{n\overline{y}_n + y_{n+1}}{n+1}\sin\theta_n \end{aligned} \tag{33}$$

In addition, the point variance of L_n, which is used in the proposed cost function, is obtained naturally and sequentially from the above equations. The variance σ_n^2 is given by:

$$\sigma_n^2 = \frac{1}{2n}\left(S_n^{xx} + S_n^{yy} - n\left(\bar{x}_n^2 + \bar{y}_n^2\right) - \sqrt{4\left(S_n^{xy} - n\bar{x}_n\bar{y}_n\right)^2 + \left(S_n^{yy} - S_n^{xx} - n\left(\bar{y}_n^2 - \bar{x}_n^2\right)\right)^2}\right)$$

and σ_{n+1}^2 with added point (x_{n+1}, y_{n+1}) becomes:

$$\sigma_{n+1}^2 = \frac{1}{2(n+1)}\left(S_n^{xx} + S_n^{yy} + \left(x_{n+1}^2 + y_{n+1}^2\right) - \frac{1}{n+1}\left((n\bar{x}_n + x_{n+1})^2 + (n\bar{y}_n + y_{n+1})^2\right) - \sqrt{4a^2 + b^2}\right) \quad (34)$$

where:

$$a = S_n^{xy} + x_{n+1}y_{n+1} - \frac{1}{n+1}(n\bar{x}_n + x_{n+1})(n\bar{y}_n + y_{n+1})$$

$$b = S_n^{yy} - S_n^{xx} + \left(y_{n+1}^2 - x_{n+1}^2\right) - \frac{1}{n+1}\left((n\bar{y}_n + y_{n+1})^2 - (n\bar{x}_n + x_{n+1})^2\right)$$

Equations 33 and 34 show that these parameters can be updated sequentially, which greatly improves the vector extraction process.

A2 Matrix Singularity

Lemma 1 *Matrices B and C of Eq. 12 are singular if and only if a single scan segment is detected, or all extracted scan segments are parallel with each other, as shown in Fig. 11c.*

Proof First, we prove the necessary condition. Given N matching pairs, matrices B and C of Eq. 13 are given by:

$$B = \sum_{k=1}^{N} \frac{l_k}{\sigma_k} \begin{bmatrix} \cos^2\theta_k \sum_{l=1}^{N_k} x_{kl} + \cos\theta_k \sin\theta_k \sum_{l=1}^{N_k} y_{kl} & -\cos^2\theta_k \sum_{l=1}^{N_k} y_{kl} + \cos\theta_k \sin\theta_k \sum_{l=1}^{N_l} x_{kl} \\ \cos\theta_k \sin\theta_k \sum_{l=1}^{N_k} x_{kl} + \sin^2\theta_k \sum_{l=1}^{N_k} y_{kl} & -\cos\theta_k \sin\theta_k \sum_{l=1}^{N_i} y_{kl} + \sin^2\theta_k \sum_{l=1}^{N_k} x_{kl} \end{bmatrix} \quad (35)$$

$$C = \sum_{k=1}^{N} \frac{l_k N_k}{\sigma_k} \begin{bmatrix} \cos^2\theta_k & \cos\theta_k \sin\theta_k \\ \cos\theta_k \sin\theta_k & \sin^2\theta_k \end{bmatrix} \quad (36)$$

When all scan vectors are parallel, θ_k is either θ or $\theta+\pi$, so that $\cos(\theta+\pi) = -\cos\theta$ and $\sin(\theta+\pi) = -\sin\theta$. Further expansion of the above yields:

$$B = \left(\sum_{k=1}^{N} \frac{l_k}{\sigma_k}\right) \cdot \begin{bmatrix} \cos^2\theta \sum_{l=1}^{N_k} x_{kl} + \cos\theta\sin\theta \sum_{l=1}^{N_k} y_{kl} & -\cos^2\theta \sum_{l=1}^{N_k} y_{kl} + \cos\theta\sin\theta \sum_{l=1}^{N_k} x_{kl} \\ \cos\theta\sin\theta \sum_{l=1}^{Nk} x_{kl} + \sin^2\theta \sum_{l=1}^{N_k} y_{kl} & -\cos\theta\sin\theta \sum_{l=1}^{N_k} y_{kl} + \sin^2\theta \sum_{j=1}^{N_k} x_{kl} \end{bmatrix} \tag{37}$$

$$= k_B \cdot B' \tag{38}$$

and:

$$C = \left(\sum_{l=1}^{N} \frac{l_k N_k}{\sigma_k}\right) \cdot \begin{bmatrix} \cos^2\theta & \cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta \end{bmatrix}$$

$$= k_C \cdot C', \tag{39}$$

where k_B and k_C are constants. Now $\det(B') = 0$ because:

$$B' = \begin{bmatrix} \cos^2\theta \sum_{l=1}^{N_k} x_{kl} + \cos\theta\sin\theta \sum_{l=1}^{N_k} y_{kl} & -\cos^2\theta \sum_{l=1}^{N_k} y_{kl} + \cos\theta\sin\theta \sum_{l=1}^{N_k} x_{kl} \\ \cos\theta\sin\theta \sum_{l=1}^{N_k} x_{kl} + \sin^2\theta \sum_{l=1}^{N_k} y_{kl} & -\cos\theta\sin\theta \sum_{l=1}^{N_k} y_{kl} + \sin^2\theta \sum_{l=1}^{N_k} x_{kl} \end{bmatrix}$$

$$= \begin{bmatrix} b_{11} & b_{12} \\ \tan\theta \cdot b_{11} & \tan\theta \cdot b_{12} \end{bmatrix}. \tag{40}$$

Similarly, $\det(C') = \cos^2\theta\sin^2\theta - \cos^2\theta\sin^2\theta = 0$. Therefore, $\det(B) = 0$ and $\det(C) = 0$, which means that matrices B and C are singular only when parallel scan vectors are obtained.

Conversely, if there are more than one scan vectors that are not parallel with other vectors, matrices B and C are not singular. When a set of all θ_k is $\{\theta_1, \theta_2, \ldots, \theta_{N_\theta}\}$, B can be arranged from Eqs. 38 and 40.

$$B = k_{B1}B_1 + k_{B2}B_2 + \ldots + k_{BN_\theta}B_{N_\theta}$$

$$= \begin{bmatrix} b_{11}^{(1)} + b_{11}^{(2)} + \ldots & b_{12}^{(1)} + b_{12}^{(2)} + \ldots \\ \tan\theta_1 b_{11}^{(1)} + \tan\theta_2 b_{11}^{(2)} + \ldots & \tan\theta_1 b_{12}^{(1)} + \tan\theta_2 b_{12}^{(2)} + \ldots \end{bmatrix} \tag{41}$$

From the definition, $\theta_1 \neq \theta_2 \neq \ldots \neq \theta_{N_\theta}$. Therefore B is not singular.

C can be arranged similarly as:

$$C = k_{C1}C_1 + k_{C2}C_2 + \ldots + k_{CN_\theta}C_{N_\theta} \tag{42}$$

$$= \begin{bmatrix} c_1\cos^2\theta_1 + c_2\cos^2\theta_2 + \ldots & c_1\cos\theta_1\sin\theta_1 + c_2\cos\theta_2\sin\theta_2 + \ldots \\ c_1\cos\theta_1\sin\theta_1 + c_2\cos\theta_2\sin\theta_2 + \ldots & c_1\sin^2\theta_1 + c_2\sin^2\theta_2 + \ldots \end{bmatrix} \tag{4.3}$$

Then $\det(C)$ is

$$\det(C) = \sum_{i \neq j} c_i c_j \cos^2\theta_i \sin^2\theta_j - \sum_{i \neq j} c_i c_j \cos\theta_i \cos\theta_j \sin\theta_i \sin\theta_j$$

From the assumption, $\det(C)$ is not zero. Therefore, C is nonsingular if nonparallel scan vectors exist. □

References

1. Anousaki, G.C., Kyriakopoulos, K.J.: Simultaneous localization and map building for mobile robot navigation. IEEE Robot. Autom. Mag. **6**(3), 42–53 (1999)
2. Arras, K.O., Siegwart, R.: Feature extraction and scene interpretation for map-based navigation and map building. In: Proceedings of the Symposium on Intelligent Systems and Advanced Manufacturing, vol. 3210, pp. 4253–4264. Mobile Robotics XII, Pittsburgh (1997)
3. Arras, K.O., Castellanos, J.A., Schilt, M., Siegwart, R.: Feature-based multi-hypothesis localization and tracking using geometric constraints. Robot. Auton. Syst. **44**(1), 41–53 (2003)
4. Ayache, N., Faugeras, O.: Hyper: a new approach for the recognition and positioning of 2D objects. IEEE Trans. Pattern Anal. Mach. Intell. **8**(1), 44–54 (1986)
5. Beveridge, J.R., Riseman, E.M.: How easy is matching 2D line models using local search?. IEEE Trans. Pattern Anal. Mach. Intell. **19**(6), 564–579 (1997)
6. Borges, G.A., Aldon, M.J.: Optimal mobile robot pose estimation using geometrical maps. IEEE Trans. Robot. Autom. **18**(1), 87–94 (2002)
7. Borges, G.A., Aldon, M.J.: Line extraction in 2D range images for mobile robotics. J Intell. Robot. Syst. **40**(1), 267–297 (2004)
8. Borges, G.A., Aldon, M.J.: Robustified estimation algorithms for mobile robot localization based on geometrical environment maps. Robot. Auton. Syst. **45**(3), 131–159 (2003)
9. Castellanos, J., Tardós, J.: Laser-based segmentation and localization for a mobile robot. In: Robotics and Manufacturing: Recent Trends in Research and Applications, vol. 6, pp. 101–108. ASME Press, New York (1996)
10. Cox, I.J.: Blanche – an experiment in guidance and navigation of an autonomous robot vehicle. IEEE Trans. Robot. Autom. **7**(2), 193–204 (1991)
11. Dudek, G., Zhang, C.: Vision-based robot localization without explicit object models. In: Proceedings of International Conference on Robotics and Automation, vol. 1, pp. 76–81. IEEE, Minneapolis (1996)
12. Einsele, T.: Localization in indoor environments using a panoramic laser range finder. Dissertation, Technischen Universität München (2000)
13. Elfes, A.: Sonar-based real-world mapping and navigation. Journal of Robotics and Automation **3**(3), 249–265 (1987)
14. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte Carlo localization: efficient position estimation for mobile robots. In: Proceedings of the National Conference on Artificial Intelligence, pp. 343–349. AAAI, Orlando (1999)
15. Haralick, R.: Propagating covariances in computer vision. In: Proceedings of the International Conference on Pattern Recognition, vol. 12, pp. 493–498. IEEE Computer Society Press, Los Alamitos (1994)
16. Lingemann, K., Nüchter, A., Hertzberg, J., Surmann, H.: High-speed laser localization for mobile robots. Robot. Auton. Syst. **51**(4), 275–296 (2005)
17. Sohn, H.J., Kim, B.K.: A robust localization algorithm for mobile robots with laser range finders. In: Proceedings of International Conference on Robotics and Automation, pp. 3545–3550. IEEE, Barcelona (2005)
18. Jensfelt, P., Kristensen, S.: Active global localization for a mobile robot using multiple hypothesis tracking. IEEE Trans. Robot. Autom. **17**(5), 748–760 (2001)
19. Lu, F., Milios, E.: Robot pose estimation in unknown environments by matching 2D range scans. J. Intell. Robot. Syst. **18**(3), 249–275 (1997)
20. Minguez, J., Lamiraux, F., Montesano, L.: Metric-based scan matching algorithms for mobile robot displacement estimation. In: Proceedings of International Conference on Robotics and Automation, pp. 3557–3563. IEEE, Barcelona (2005)

21. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: Proceedings of the National Conference on Artificial Intelligence, vol. 18, pp. 593–598. AAAI, Orlando (2003)
22. Mota, J.G., Ribeiro, M.I.: Mobile robot localisation on reconstructed 3D models. J. Robot. Auton. Syst. **31**(1), 11–30 (2000)
23. Nguyen, V., Martinelli, A., Tomatis, N., Siegwart, R.: A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems, vol. 2005, pp. 1929–1934. IEEE, Alberta (2005)
24. Pavlidis, T., Horowitz, S.: Segmentation of plane curves. IEEE Trans. Comput. **23**, 860–870 (1974)
25. Pfister, S., Roumeliotis, S., Burdick, J.: Weighted line fitting algorithms for mobile robot map building and efficient data representation. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 1304–1311. IEEE, Taipei (2003)
26. Pfister, S.: Algorithms for mobile robot localization and mapping, incorporating detailed noise modeling and multi-scale feature extraction. Dissertation, California Institute of Technology (2006)
27. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling, pp. 145–152. IEEE, Quebec (2001)
28. Siegwart, R., Nourbakhsh, I.: Introduction to Autonomous Mobile Robots. MIT Press, Cambridge (2004)
29. Weber, J., Jörg, K.W., Puttkamer, E.: APR-Global scan matching using anchor point relationships. In: Proceedings of the 6th International Conference on Intelligent Autonomous Systems, pp. 471–478. IOS press, Venice (2000)
30. Weiß, G., Wetzler, C., Puttkamer, E.: Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In: Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems, vol. 1, pp. 595–601. IEEE, Munich (1994)
31. Zezhong, X., Ronghua, L., Jilin, L.: Global localization based on corner point. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, vol. 2, pp. 843–847. IEEE, Kobe (2003)
32. Zhang, L., Ghosh, B.K.: Line segment based map building and localization using 2D laser rangefinder. In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 1, pp. 2538–2543. IEEE, San Francisco (2000)