Research Article

Novel Point-to-Point Scan Matching Algorithm Based on Cross-Correlation

Jaromir Konecny,[1] Michal Prauzek,[1] Pavel Kromer,[2] and Petr Musilek[3]

[1]*Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, 17. Listopadu 2172/15, 70833 Ostrava, Czech Republic*
[2]*Department of Computer Science, VSB-Technical University of Ostrava, 17. Listopadu 2172/15, 70833 Ostrava, Czech Republic*
[3]*Department of Electrical and Computer Engineering, Faculty of Engineering, University of Alberta, 9107-116 Street, Edmonton, AB, Canada T6G 2V4*

Correspondence should be addressed to Jaromir Konecny; jaromir.konecny@vsb.cz

The localization of mobile robots in outdoor and indoor environments is a complex issue. Many sophisticated approaches, based on various types of sensory inputs and different computational concepts, are used to accomplish this task. However, many of the most efficient methods for mobile robot localization suffer from high computational costs and/or the need for high resolution sensory inputs. Scan cross-correlation is a traditional approach that can be, in special cases, used to match temporally aligned scans of robot environment. This work proposes a set of novel modifications to the cross-correlation method that extend its capability beyond these special cases to general scan matching and mitigate its computational costs so that it is usable in practical settings. The properties and validity of the proposed approach are in this study illustrated on a number of computational experiments.

1. Introduction

Accurate and efficient positioning and localization is a fundamental problem of mobile robotics. It involves estimation of robots' position relative to a map of an environment [1]. To accomplish this task, mobile robots adopt two high-level localization approaches. They can determine their position by receiving signals from beacons, such as in the case of fingerprinting algorithms [2] or employ various sensory subsystems that inform them about their vicinity [3]. Devices commonly used for beaconless localization are wheel sensors (odometers) and ultrasonic and optical rangefinders [4].

Optical rangefinders perform 2D laser scans of robot surroundings and provide data with high resolution and at high sampling rates. In general, the processing of such data is computationally expensive and usually requires massive computing resources [5]. On the contrary, control systems of mobile robots are usually low-consumption embedded devices with limited resources, low performance, and small memory. Therefore, there is a clear need for innovative laser

scan processing methods with a good trade-off between accuracy and computational complexity. In this paper, a novel cross-correlation-based scan matching method suitable for low-performance microcontrollers is proposed and evaluated. It is an efficient cloud point-matching algorithm that can be in mobile robots instead of the traditional methods such as the Iterative Closest Point (ICP) [6], Cox [7], complete line segment (CLS) [8], Normal Distributions Transform (NDT) [9], Perimeter-Based Polar Scan Matching (PB-PSM) [10], and, for example, pIC [11] algorithm.

The novel cross-correlation-based [12] scan matching method, proposed in this work, uses laser scans obtained by optical rangefinder to solve the simultaneous localization and mapping (SLAM) problem [13] and to determine robot position in an unknown environment. The proposed algorithm has been implemented in C# and evaluated in a series of computational experiments involving a realistic mobile robot platform equipped with a specific optical rangefinder (SICK LMS 100 [14]). The accuracy and performance of the proposed method have been compared to a standard

scan matching algorithm, ICP, and found better in terms of processing time and the accuracy of estimated position.

The rest of this paper is organized as follows. The scan matching problem, a general classification of scan matching methods, and the definition of selected baseline scan matching methods is provided in Section 2. Section 3 gives a brief overview of recent related work and relevant approaches. An efficient and robust scan matching algorithm, based on the cross-correlation of rasterized LiDAR scans, is proposed in Section 4. Section 5 describes the experiments conducted in order to verify the approach and to assess its properties. Finally, conclusions are drawn and future work is outlined in Section 6.

2. Scan Matching

Informally, scan matching (cloud point matching) is a general procedure that aims at aligning current scan of an environment with a reference scan [15]. Many methods, based on various principles and different formal approaches, have been proposed for scan matching in the past. However, most of them suffer from high computational costs [16] and only a limited ability to work efficiently in different environments [17] (e.g., the method described in [18] requires an environment with perpendicular walls).

Scan matching-based robot localization methods utilize information about the distance between the device and the nearest obstacle. This information can be obtained with high accuracy using a laser range finder (LiDAR) [14]. In these devices, a measuring beam is often swept in one axis and provides the information about the distance to the nearest obstacle at every measured angle. Common LiDARs provide approximately 10–50 such scans per second. Each scan contains information about the distances to the nearest obstacle within a plane in front of the device (2D LiDAR). A typical LiDAR, such as the SICK LMS 100, has a measurement range of 270° with an angular resolution 0.25° [14]. Effective measurement distance ranges from several meters to tens of meters, depending on sensor types and properties. Besides traditional 2D LiDARs, devices able to provide 3D scans of their environment are becoming increasingly popular [19].

Thanks to their favourable properties, 2D LiDARs have become considerably popular in robotics. There are several methods, based on various heuristics and principles, that can be employed to determine the position of a robot in an environment. Methods that align current scan of an environment with a reference scan or with a map are called scan matching methods.

Scan matching methods can be divided into two large groups. *Conventional scan matching* methods use the apparatus of classical mathematics, while *probabilistic scan matching* methods evaluate the likelihood of a robot being at certain place. Typical examples of conventional and probabilistic methods are the Iterative Closest Point algorithm [6] and the Normal Distribution Transform algorithm [20], respectively.

Another classification of scan matching procedures is based on the way scan data is being processed. The *point-to-point scan matching* strategies process all individual points

in environment scans. They provide localization with a high accuracy but suffer from high computational costs. However, they are very well usable in both complex and nonstructured environments. The *feature-to-feature* methods extract from the scans higher-level features before the actual matching and localization take place. The extracted features can be diverse. They usually include basic geometric shapes such as lines, arcs, edges, polygons, and, for example, 3D features. These algorithms have lower computational cost of the matching phase but can operate only in sufficiently feature-rich (i.e., structured) environments. They perform well in buildings with well-structured environmental elements consisting of large, flat surfaces and regular, geometric shapes. In the following, the standard scan matching methods from both categories are summarized.

2.1. Point-Based Scan Matching Methods

2.1.1. Iterative Closest Point Algorithm. The ICP is an iterative algorithm that looks for the pairs of closest points in a pair of environment scans. An affine transformation, **T**, that makes projection of one point to another is calculated between two different scans, A and B. The algorithm minimizes a loss function, $f(p_x, p_y, \omega)$, defined as

$$f\left(p_x, p_y, \omega\right) = \sum_{i=1}^{n} \left\| \mathbf{T}\left(a_i, \left(p_x, p_y, \omega\right)\right) - b_{\vartheta(i)} \right\|^2, \quad (1)$$

where **T** is the affine transformation, p_x and p_y are translations in the direction of the x- and y-axes, respectively, ω is rotation, and ϑ is a function that finds in the second scan, B, index of a point that is closest to the point from the original scan with index i.

The result of the minimization is a three-element vector that represents the translations in x- and y-axes and the rotation of the test scan with respect to the reference scan [21]. The ICP algorithm can be summarized as follows [6]:

(1) Preprocessing: removal of the remote points.

(2) Assignment: finding pairs of the closest points (the first point is from the reference scan; the second point is from the test scan).

(3) Rejection: removal of the pairs with the long distance.

(4) The loss function calculation: equation (1).

(5) The loss function minimization: iterative process (e.g., Newton method or Lorentzian estimator [22]).

Loss function minimization is the key part of the algorithm. Minimization methods with good trade-off between accuracy (i.e., the ability to find good transformations) and computational costs are required for mobile robots equipped with energy and resource constrained microcontrollers.

2.1.2. Histogram Correlation. Correlation methods based on histograms, such as the Extended Histogram Matching algorithm, can be used to accomplish scan matching as well [12]. However, traditional correlation can be applied only for scans that differ in rotation only. For two arbitrary scans that differ

in rotation, the function $r(\Phi)$ that represents the laser scan differs in shift in the Φ axis. If the scans differ in both, rotation and translation, the function $r(\Phi)$ differs in distribution and the algorithms may produce misleading results (i.e., wrong matching).

Histogram-based correlation methods therefore use histograms, including the angle histogram [12], to determine the rotation and translation of the matched scans. They compute a histogram of the angles between every pair of points, $r(\Phi_i)$ and $r(\Phi_{(i+1)})$, measured in the scan. The function, obtained in this way, is invariant towards displacement. The x- and y-axis histograms show the distributions of points scanned in these two directions. Histogram-based scan correlation is described in detail in [12].

2.2. Feature-Based Methods. A typical example of feature-based scan matching methods is the complete line segment (CLS) algorithm that compares complete line segments, extracted from two different scans. It can be also used to match a scan with a reference map of environment. This method has been successfully applied to scan matching [8] and for SLAM [23].

The CLS extracts from each LiDAR scan line segments that represent high-level real-world objects found in robot environment. The lines can be either complete or incomplete. An incomplete line is a sign of one object occluded by another. A complete line segment, on the other hand, describes a real-world object in plain view of the robot. The algorithm is especially concerned with complete line segments. It assumes that a complete line segment has an unique Euclidean length within the environment. The map of an environment is made up of a set of lines with defined start and end points and corresponding length. However, the lines can be defined also by their center point, orientation, and length.

Line comparisons (i.e., scan matching) are performed in CLS using the length of line segments, relative position of their center points, and their relative rotation. Let us consider two complete line segments L and G. L is a segment from a local map, and G is a segment from a global map. Together, they form a pair. The CLS works in the following way. For all line segments from the local map, L_i, one by one, it builds a set of segments from the global map, G_m, with similar length. Then, it calculates for each pair L_i and $G_i \in G_m$ the relative position of their centers and their relative rotation. The segment, i, is matched if the following condition is satisfied:

$$\left| L_s L_{si} \right| \doteq \left| G_s G_{si} \right| \wedge L_\alpha - G_\alpha \doteq L_{\alpha i} - G_{\alpha i} \wedge \left| L_i \right| \doteq \left| G_i \right|, \quad (2)$$

where the midpoint of a segment is denoted by subscript s and the relative rotation by subscript α. The more the segments meet the condition given by (2), the greater the credibility of the test match is.

If the test match contains at least two corresponding pairs, it is possible to calculate the angle, ω, and the displacement parameters, p_x and p_y, respectively. The angle, ω, is calculated from two pairs of complete line segments as a difference between an orientation vector, created from the midpoints of the local line segments, L_{si} and L_{sj}, and a vector created

from the midpoints of the global line segments, G_{si} and G_{sj}. It is possible to use any of those two pairs of segments for the calculation. The displacement parameters are computed using

$$p_x = x_G - \left(x_C \cos \omega - y_C \sin \omega \right),$$
$$p_y = y_G - \left(y_C \cos \omega + x_C \sin \omega \right). \quad (3)$$

The scan matching procedure, proposed in this work, is a novel point-based method developed especially for the segment of energy and power constrained devices such as mobile robots. In the next section, we briefly summarize relevant related approaches.

3. Related Work

A SLAM method based on stereo vision and the ICP algorithm has been described in [24]. SLAM method based on laser scan matching has been introduced in [25], where authors use polar coordinates for scan matching. The combination of the ICP algorithm and correlation histogram is used in [26] for large scale SLAM. In [27] the SLAM method based on entropy is presented. In [28] authors propose a beam selection method. The laser sensor beam is filtered and only the most important beams are used for SLAM. The representative of a multiagent approach is presented in [29].

6-DoF low dimensionality SLAM (L-SLAM) is introduced in [19]. Authors use 3D kinematic model instead of 2D. The particle filter and Kalman filter are used in that SLAM. The alternative approach is introduced in [30] that uses Extended Kalman Filter (EKF). Authors also present the SLAM comparison.

Another frequently used approach is based on the extraction of geometric primitives. For example, these primitives can take form of line segments [8] or more complex 3D segments [31]. In [31] the authors use 3D landmarks for feature-based SLAM. Another example of feature-based SLAM method is in [32], where the linear group algorithm (LGA) and stereo vision is used for SLAM. In [33] authors deal with a kidnap problem. They use a upward-looking camera for a first pose estimation.

Variety of additional information can be included in the maps. Those pieces of information can be used in subsequent analysis of explored area. The mobile robot that explores a waste rock is described in [34]. The concentration of a carbon monoxide and methane is measured and collected. The global positioning system (GPS) and online maps are used for localization. Another research is referred to in [35], where a mobile robotic device for mapping a distribution of a gas is presented.

In [36] the wireless node localization is proposed. This method is suitable for indoor use, while GPS signal is not present. Monte Carlo localization is used for wireless node identification. The localized nodes can be afterwards used for backward localization.

The following section describes the proposed novel cross-correlation-based scan matching method in detail. It provides an efficient an accurate algorithm for evaluating the degree

of similarity between two laser scans, called correlation coefficient. The correlation coefficient is in this work a number that represents the overlay of two LiDAR scans. The larger the correlation coefficient is, the better the match is obtained between the investigated LiDAR scans. The calculation of the correlation coefficient is crucial for computational costs of the scan matching procedure. In this approach, the 2D point cloud, generated by the sensor, is transformed into a lower-resolution raster and then used to evaluate how much the borders of the investigated scans collide.

4. Point-to-Point Scan Matching Algorithm Based on Cross-Correlation

In this work, we propose a robot localization strategy using a novel point-to-point scan matching algorithm based on cross-correlation. The proposed method has low computational requirements and high accuracy and is therefore suitable for the use with embedded devices that are frequently found in mobile robot platforms. The cross-correlation is in this approach used to determine relative translation and rotation of consecutive LiDAR scans performed by a moving robot. Each LiDAR scan can be in this context understood as a momentary snapshot of a floor plan of a room (more general, environment) where the robot is located. The scans have usually angular resolution of $1°$ and cover the entire neighborhood of the robot (i.e., $360°$). They contain for each measured angle a number of points that indicate the distance between the robot and nearest obstacle in the corresponding direction.

Intuitively, two LiDAR scans performed in the same environment shortly after each other will be similar. The proposed scan matching approach finds an affine transformation vector, $\mathbf{p}$, that is the best projection between an actual and a reference LiDAR scan. The transformation vector consists of three elements, transformation parameters, that correspond to translations, p_x and p_y, and rotation, ω.

Consider a set of all possible affine transformation vectors, Ψ, and a vector, $\mathbf{p}$:

$$\mathbf{p} = \left(p_x, p_y, \omega \right), \quad \mathbf{p} \in \Psi, \tag{4}$$

where p_x and p_y are the translation and ω is the rotation.

An affine transformation, $\mathbf{T}$, based on a parameter vector, $\mathbf{p} = (p_x, p_y, \omega)$, is defined by

$$\mathbf{T} : \begin{pmatrix} b_x \\ b_y \end{pmatrix} = \begin{pmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{pmatrix} \cdot \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \end{pmatrix}, \tag{5}$$

where (a_x, a_y) and (b_x, b_y) are the coordinates of a point, (x, y), in two LiDAR scans, A and B, respectively.

There is a handful of methods, based on different formal approaches and designed for various applications, that can find the parameter vector for the affine transformation between A and B. Some of them are summarized in Section 2. In the following, a novel cross-correlation-based method suitable for embedded microcontrollers is proposed. The method is first defined for two scans that differ only in translation and then extended to match scans when both, translation and rotation, are performed.
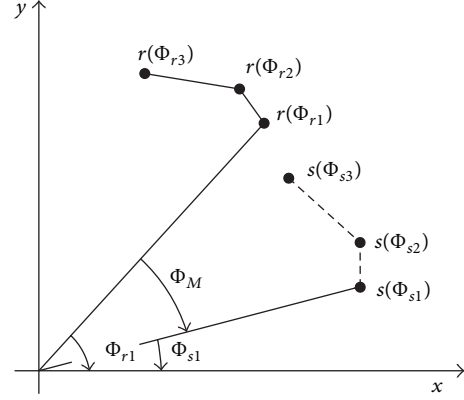

FIGURE 1: Two scans with different rotation (three points only).

4.1. Cross-Correlation of Rotated-Only LiDAR Scans.
Consider two LiDAR scans, A and B, that differ only by rotation. Assume that they were captured in a sufficiently indented environment so that the functions $r(\Phi)$ and $s(\Phi)$, representing the scans A and B, respectively, as a function of rotation angle, Φ, have a period of 2π. An example of this scenario is illustrated in Figure 1. The figure shows three points, 1, 2, and 3, in scans A and B, respectively. Each point, i, is in scan A represented by $r(\Phi_{ri})$ and in scan B by $s(\Phi_{si})$. The cross-correlation of two rotated-only 2D scans, A and B, can be then evaluated using [37]

$$(r * s)(\Phi) = \int_0^{2\pi} r(\tau) \cdot s(\Phi + \tau) \cdot d\tau. \tag{6}$$

Formula (6) is a function of scan similarity that depends on the angle, Φ, only. The rotation between the matched scans A and B, Φ_M, is then simply calculated by

$$\Phi_M = \arg \max_{\Phi \in [0, 2\pi]} \left\{ (r * s)(\Phi) \right\}. \tag{7}$$

The value of Φ_M can be easily obtained from (7) using a single scan along the domain of Φ at selected angular resolution. This intuitive approach requires only a single program loop and is computationally acceptable even for low-power embedded devices. Unfortunately, the cross-correlation problem is significantly more complex for a general case of two LiDAR scans that differ in both rotation and translation.

4.2. Cross-Correlation of Arbitrary LiDAR Scans.
An example of two LiDAR scans, A and B, differing by rotation and translation at the same time, is shown in Figure 2. A modified, computationally efficient, cross-correlation algorithm needs to be introduced to obtain both rotation and translation in order to match such arbitrary LiDAR scans.

Let us define modified correlation as a function of three parameters, $(r * s)(\mathbf{p})$. In order to evaluate $(r * s)$, several steps need to be carried out. First, it is appropriate to convert the scans into Cartesian coordinates to calculate the translation.
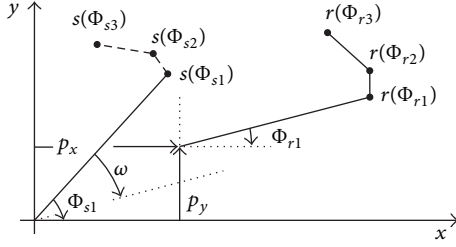
FIGURE 2: Two scans with different rotation and translation. Only three points, 1, 2, and 3, are displayed.

The conversion from polar to Cartesian coordinates is defined by

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r(\Phi) \cdot \cos \Phi \\ r(\Phi) \cdot \sin \Phi \end{pmatrix}. \tag{8}$$

Next, we assume an abstract operation, $\odot$, that will replace the conventional correlation so that the value of the *modified correlation function*, $(r * s)(\mathbf{p})$, will reflect the degree of alignment of two LiDAR scans for each transformation vector $\mathbf{p} = (p_x, p_y, \omega)$:

$$\begin{aligned}
&(r * s)\left(p_x, p_y, \omega\right) \\
&= \begin{pmatrix} r(\Phi) \cdot \cos \Phi \\ r(\Phi) \cdot \sin \Phi \end{pmatrix} \\
&\quad \odot \begin{pmatrix} s(\Phi + \omega) \cdot \cos(\Phi + \omega) + p_x \\ s(\Phi + \omega) \cdot \sin(\Phi + \omega) + p_y \end{pmatrix}.
\end{aligned} \tag{9}$$

The result of this operation is a positive real number called *correlation coefficient*. The correlation coefficient corresponds to the degree of similarity between the LiDAR scans modified by $\mathbf{p}$. The best transformation vector, $\mathbf{p}_M$, describing the rotation and translation between A and B most accurately, can be obtained from (9) using

$$\mathbf{p}_M = \arg \max_{(x_M, y_M, \omega_M)} \left\{ (r * s)\left(x_M, y_M, \omega_M\right) \right\}. \tag{10}$$

When finding $\mathbf{p}_M$, an exhaustive search along the domains of x_M, y_M, and ω_M is performed to solve (10). Such exhaustive search or any suitable real parameter optimization method that can be employed to find an optimum solution to (10) requires many evaluations of $(r * s)(p_x, p_y, \omega)$. Apparently, a reasonable and computationally lightweight $\odot$ is crucial for practical determination of $\mathbf{p}_M$.

4.3. Correlation Coefficient. A computationally efficient but sensitive and accurate method for the evaluation of (9) is the key element of the proposed scan matching approach. In this section, we discuss computational methods for the evaluation of (9) and devise an algorithm suitable for practical deployment in the field of mobile robot localization.
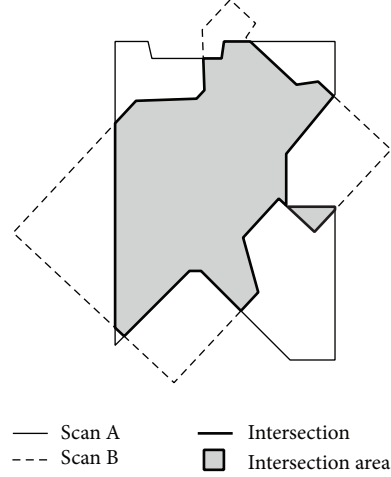

FIGURE 3: Intersection representation of correlation coefficient.

4.3.1. Simple Correlation Coefficient Evaluation. A basic, intuitive, representation of the correlation coefficient can be based on a sum of the functional representations of two LiDAR scans, A and B, $r(\Phi)$ and $s(\Phi)$, respectively. This is in the general case generally defined by (6). In practice, the scans are sampled from $r(\Phi)$ and $s(\Phi)$ at certain angular resolution and the formula is discretized as

$$(r * s)(n) = \sum_{i=0}^{n} r(i) s(n + i). \tag{11}$$

The calculation of the best transformation vector, defined by (10), is a computationally intensive task. An exhaustive search a cross the domains of x_M, y_M, and ω_M requires three nested loops and triggers a very large number of $\odot$ evaluations. This operator can be numerically expressed in different ways and its choice affects the efficiency of the scan matching procedure substantially.

Figure 3 displays two mutually shifted and rotated LiDAR scans. One way to define $\odot$ in this arbitrary case is to measure the overlapping area of both matched scans. The most accurate transformation vector, $\mathbf{p}$, is for two matched scans obtained when the area of the intersection is maximized.

The scans are in general polygons and thus this approach requires finding the intersection of two general polygons. This can be accomplished, for example, by the Weiler-Atherton algorithm [38]. However, one laser scan consists of hundreds to thousands of values and intersection calculation is very computationally expensive. In our experiments, it took more than 30 seconds in laboratory conditions. Moreover, finding the intersection is only the first step and the area of the overlapping region has to be computed as well. This naïve approach is clearly too computationally expensive and infeasible for the use in microcontrollers of mobile robots.

4.3.2. Raster-Based Correlation Coefficient Evaluation. A more efficient way to obtain the correlation coefficient is via rasterization of the 2D scans. The overlapping area (i.e., the correlation coefficient) is then the number of raster cells

☐ Scan A	◼ Intersection
◼ Scan B	- - - Scan B
— Scan A	

(a)

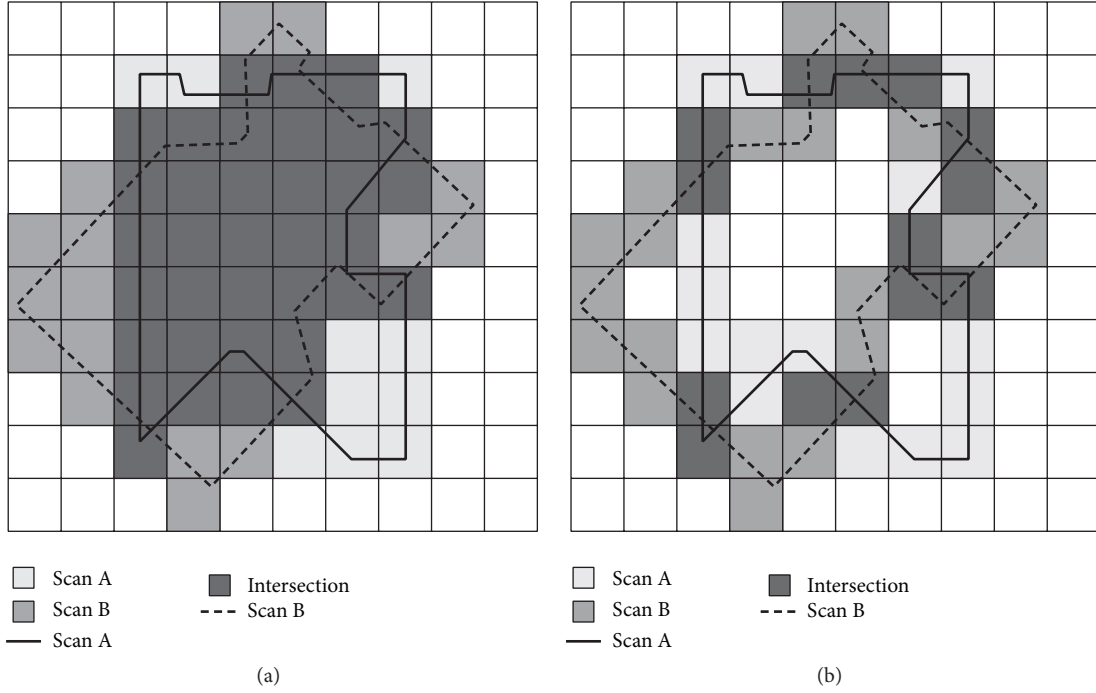|	
☐ Scan A	◼ Intersection
◼ Scan B	- - - Scan B
— Scan A	

(b)

FIGURE 4: Correlation coefficient representation in raster. (a) Area representation, (b) boundary representation.

that occur in both scans at the same time. This is illustrated in Figure 4(a). However, this approach still requires a determination whether each cell is inside or outside of the polygons. That requires time and resources and, as shown in the following, is not really necessary for the evaluation of a suitable correlation coefficient.

The proposed algorithm avoids the computation of the overlapping area altogether. The correlation coefficient is computed solely on the basis of the boundaries of the polygons obtained from scan A and scan B, respectively. In order compute the intersection of rasterized polygons accurately, the matched LiDAR scans have to describe the environment with sufficient detail and the data has to be dense. A concept of cell weight is defined to adjust the evaluation of the correlation coefficient. Cell weight, w_a, is the count of measured points that belong to a particular cell, a. The experiments show that the use of cell weight can mitigate the error introduced by bad measurements and outliers by suppressing the influence of cells with little measured points. A visual example of this approach is displayed in Figure 4(b). Needless to say, this method is the least computationally expensive algorithm for correlation coefficient calculation. However, it requires that the matched LiDAR scans describe the environment with sufficient detail and the data is dense.

4.4. An Efficient Implementation of Correlation Coefficient Evaluation. The raster-based approach to the correlation coefficient evaluation requires an efficient implementation for a practical deployment in low-power embedded devices. The degree of similarity between two LiDAR scans is in the examples, given in Figure 4, evaluated by scanning two 10×10 rasters of weighted cells. A brute-force comparison of

both rasters requires 100 steps. However, the actual polygons interfere only with a small number of raster cells (a total of 29 in the example in Figure 4). In order to achieve a memory efficient LiDAR scan storage and a faster calculation of the correlation coefficient, a sparse representation of the rasters is adopted. The representation, utilizing a sparse matrix as the data structure to store the matched scans, is illustrated in Figure 5. An evaluation of the correlation coefficient requires with this data structure a significantly lower number of steps equal to the total number of nonoverlapping raster cells occupied by both rasterized scans.

The optimum size of the cells in the raster, suitable for practical use by mobile robots, was determined empirically based on the initial experiments. The experiments have shown that 200 mm is the optimum size of a cell that achieves a good trade-off between accuracy and computational costs. Larger cell sizes lead to wrong scan matching results (low accuracy) and smaller cell sizes increase the required computational effort and compromise the robustness of the algorithm.

The complete scan matching procedure, proposed in this paper, is implemented using the following simple steps:

(1) Fetch next cell from the reference sparse matrix, A.

(2) If the current scan, B, includes on the corresponding position, x, y, a nonempty cell, increase the correlation coefficient.

(3) Go back to (1), until all cells in A are processed.

The update of the correlation coefficient, v, performed in step (2) of the scan matching procedure outlined above, can be implemented using a number of distinct strategies.

Scan A (x, y):

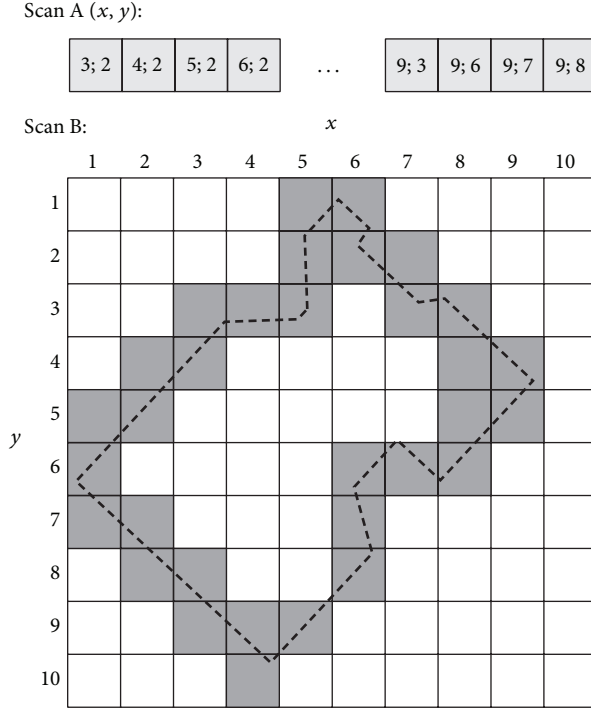| 3; 2 | 4; 2 | 5; 2 | 6; 2 | ... | 9; 3 | 9; 6 | 9; 7 | 9; 8 |

FIGURE 5: Sparse matrix representation; scan A is represented by a set of occupied cells; numbers in cells represent coordinates (x, y) in scan raster; scan B is represented by complete scan raster.

Two approaches have been found suitable by computational experiments. The correlation coefficient can be increased by the arithmetic mean of the weight of the matched cells, a and b:

$$v_{(k+1)} = v_{(k)} + w_a + w_b. \tag{12}$$

Another correlation coefficient update method uses the square root of the product of the weights of both cells:

$$v_{(k+1)} = v_{(k)} + \sqrt{w_a \cdot w_b}. \tag{13}$$

The mobile robot localization strategy, based on the proposed cross-correlation procedure, continuously compares two LiDAR scans. This process can be executed in parallel to exploit the capabilities of modern multicore processors. The data-parallel algorithm is executed by multiple threads of execution at the same time. Each thread seeks for the best transformation vector, $\mathbf{p}_p$, in a section of the rotation interval, $\Psi_p \in \Psi$. The complete rotation interval, Ψ, is divided into p disjoint intervals, $\Psi_1, \Psi_2, \ldots, \Psi_p$, so that

$$\Psi_1, \Psi_{p2}, \ldots, \Psi_p \subset \Psi,$$

$$\Psi_1 \cup \Psi_2 \cup \cdots \cup \Psi_p = \Psi, \tag{14}$$

$$\Psi_i \cap \Psi_j = \emptyset, \quad \text{for } i \neq j.$$

One transformation vector, $\mathbf{p}_p$ is found for each rotation interval subset, Ψ_p. The cross-correlation algorithm is then applied again to find the most appropriate transformation
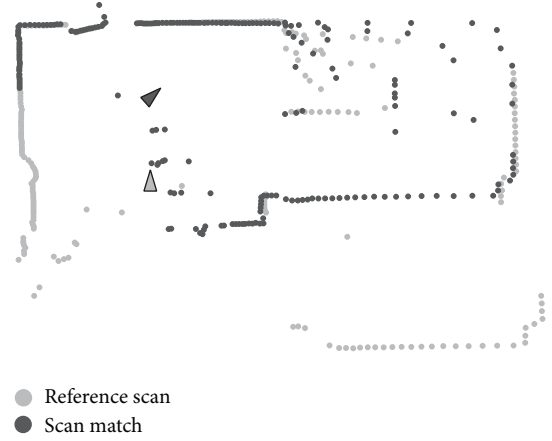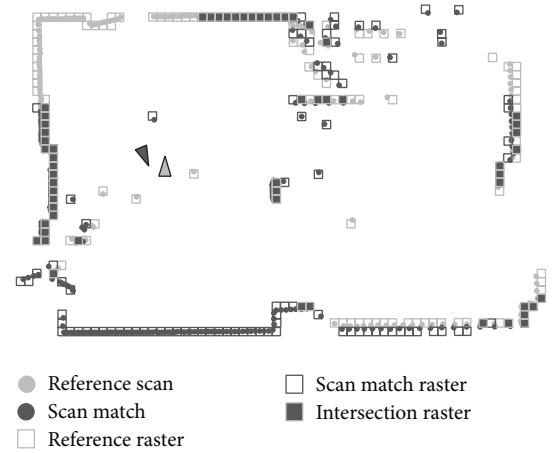

FIGURE 6: Scan matching verification 1.

FIGURE 7: Raster map representation.

vector, $\mathbf{p} = (x_M, y_M, \omega_M)$, from all locally matched transformation vectors, $\mathbf{p}_p$.

5. Experiments and Results

In order to verify the proposed scan matching algorithm, a test and simulation platform was conceived. The application was implemented in C# and linked to a LMS 100 laser sensor to allow an online testing and verification. The application provides a user interface that displays the LiDAR scans and provides a well-designed environment for computational experiments and their evaluation. Figures 6, 7, and 8 are screen captures, taken directly from the application.

5.1. Basic Verification. A verification of the proposed cross-correlation method in a real-world experiment is shown in Figure 6. The screen capture shows a test scan, displayed by the dark-gray dots, that is visually well aligned with a reference scan, outlined by the light-gray dots. An auxiliary numerical *similarity degree*, λ, of two rasterized scans, based on their mutual overlap, is used to easily evaluate the results of the scan matching process. It is defined as the ratio of
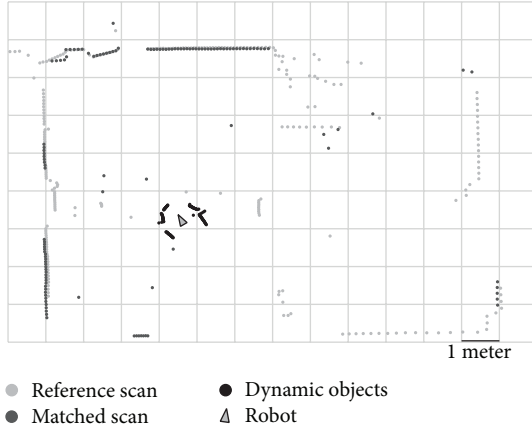
Reference scan · Dynamic objects
Matched scan · Robot

FIGURE 8: Dynamic objects robustness.

TABLE 1: The dynamic objects robustness verification in zero position.

x (mm)	y (mm)	ω (°)	λ (%)	x (mm)	y (mm)	ω (°)	λ (%)
5 obstacles 2 m				5 obstacles 1 m			
0	0	0	77	0	0	0	53
0	0	0	69	0	0	0	53
0	0	0	70	0	0	0	56
0	0	0	76	0	0	0	61
0	0	0	72	0	0	0	59
5 obstacles 1.5 m				5 obstacles 0.5 m			
0	0	0	67	0	0	0	34
0	0	0	67	0	0	0	21
0	0	0	64	3000	3200	28	11
0	0	0	65	3000	3200	38	7
0	0	0	63	0	0	0	27

TABLE 2: Dynamic objects robustness verification in position [630, 800].

x (mm)	y (mm)	ω (°)	λ (%)	x (mm)	y (mm)	ω (°)	λ (%)
5 obstacles 2 m				5 obstacles 1 m			
600	−800	30	44	600	−800	30	33
600	−800	30	45	600	−800	30	29
600	−800	30	44	600	−800	30	30
600	−800	30	40	600	−800	30	25
600	−800	30	38	600	−800	30	23
5 obstacles 1.5 m				5 obstacles 0.5 m			
600	−800	30	44	600	−800	30	26
600	−800	30	41	1000	−400	38	21
600	−800	30	37	400	3200	34	18
600	−800	30	35	600	−800	30	38
600	−800	30	31	600	−800	30	23

the cells that occur in both matched scans and the total number of cells in the raster.

The similarity degree of the two scans, displayed in Figure 6, is equal to 38%. Even though the corresponding transformation vector, $\mathbf{p} = (0, 2000, −48°)$, matches the real translation and rotation of the mobile robot, the experiments clearly confirm that the proposed approach is robust and copes well even with large changes in translation and rotation. Another example of two correctly matched scans is shown in Figure 7. This figure also demonstrates another feature of the test environment, the reference rasters of both scans, and their intersection. The transformation vector of this example is $\mathbf{p} = (−600, 400, −158°)$ and the degree of similarity, λ, is in this case equal to 46%.

5.2. Robustness to Dynamic Objects. An important advantage of the proposed algorithm is its robustness towards dynamic objects (e.g., animals, people, and vehicles) in the environment. More specifically, dynamic objects are previously nonexisting entities that had appeared in the environment between the reference scan and the current scan. The performed experiment demonstrates the ability of the proposed algorithm to deal with objects that appear in the vicinity of the robot. It examines how the number of dynamic objects and their location relative to the sensor affect the results of position estimation. The direction and speed of the objects is not considered in this experiment because the main impact on the position estimation is caused by the obstruction of the sensor's angular field of view.

The experiment started with a reference scan of the environment without any dynamic objects. Then, five obstacles were distributed around the sensor randomly in a 2-meter distance to simulate a dynamic scene with moving objects. The distance between the objects and the sensor was then reduced to 1.5, 1.0, and 0.5 meters, respectively. After every change, a LiDAR scan was performed and matched with the original reference scan. The results of this matching are displayed in Table 1. It shows that the presence of multiple moving objects does not affect the results of the matching algorithm until the distance dropped to 0.5 meters. Although

the scan similarity degree, λ, changed with every scene change, zero translation and rotation were reported. This is a correct result since the position of the sensor did not change during the experiment. When the distance between the sensor and the moving objects dropped to 0.5 meters, the visibility of the environment fell below an acceptable threshold and an incorrect translation and rotation were calculated. This also illustrates another property of the proposed scan matching algorithm: if the difference between the reference scan and the current scan is above certain threshold, the scan matching ends with an incorrect result. The higher the speed of the robot is, the faster this happens.

Table 2 shows the results of an extended version of this experiment. The sensor was again surrounded by a set of five objects, approaching its original location. However, it was relocated between the LiDAR scans this time. Also in this case, the proposed scan matching procedure worked correctly until the distance between the sensor and the moving objects fell to 0.5 meters. The experiment is illustrated by a screen from the test environment shown in Figure 8.

TABLE 3: Time cost comparison.

	B_D (ms)	$Q_{0.25}$ (ms)	$Q_{0.50}$ (ms)	$Q_{0.75}$ (ms)	B_U (ms)
Corr.	10	16	25	30	51
ICP	207	299	333	369	474
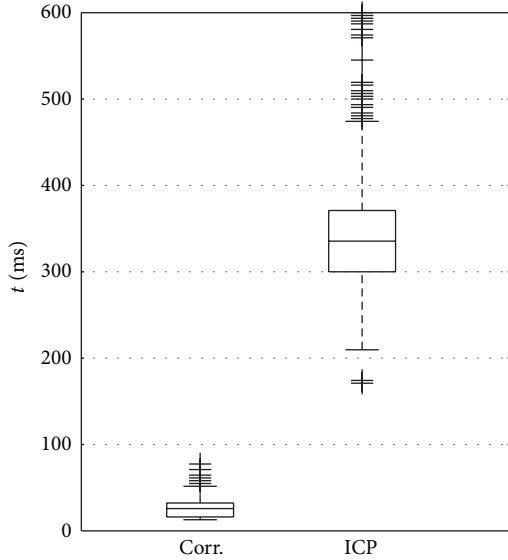
FIGURE 9: Time consumption comparison.

TABLE 4: Absolute and relative error comparison.

	Corr.		ICP	
	ϵ (mm)	δ (%)	ϵ (mm)	δ (%)
B_D	−100	−4.23	−16.99	−4.00
$Q_{0.25}$	−50	−0.99	−4.00	−1.11
$Q_{0.50}$	0	0	0	0
$Q_{0.75}$	50	1.41	5.01	0.85
B_U	100	4.92	12.39	3.61

5.3. Computational Costs of the Algorithm. In order to assess the practical time requirements of the proposed algorithm and to compare them with a standard scan matching method, ICP, a large number of LiDAR measurements and scan matchings were performed by both methods. The experiments were performed on a laboratory laptop with an Intel® Core™ i5-3360M CPU @ 2.80 GHz and 8 GiB of main memory. The results of 1,000 measurements and matchings are displayed as a box plot in Figure 9 and statistically evaluated in Table 3. The columns in Table 3 represent the lower and upper adjacent values (B_D and B_U) and first ($Q_{0.25}$), second ($Q_{0.50}$), and third quartile ($Q_{0.75}$), respectively. It shows that the typical execution time of the proposed algorithm is between 16 ms and 30 ms. That makes it approximately 10 times faster than the ICP method. This comparison shows that it is suitable for an online usage.

5.4. Measurement Error. The accuracy of the localization, based on the proposed cross-correlation method, was compared with the ICP localization in a series of experiments. In general, absolute error, defined by

$$\epsilon = X_m - X_{actual}, \tag{15}$$

and relative error, given by

$$\delta = \frac{X_m - X_{actual}}{X_{actual}}, \tag{16}$$

were evaluated. In the above, X_m is a measured value of an arbitrary variable and X_{actual} is an actual quantity of this

variable. The accuracy of the proposed method was compared to the ICP when determining the translation between two LiDAR scan.

The experiment started by taking a reference scan of the environment in an initial location. Then, the robot was moved ahead in a sequence of 5 cm steps. 100 scans were executed at each step and matched to the reference scan. The distance from the initial location ranged up to 8 m. The cell size in the correlation algorithm was set to 200 mm. The actual translation, X_{actual}, was measured manually by an accurate tape meter. The results of this experiment are shown in Figure 10 and summarized in Table 4. Together, they show that the upper adjacent value of the absolute error, which is taken as a representative value of this measure, was ±100 mm for the cross-correlation method and between −16.99 mm and 12.39 mm for the ICP algorithm.

Figure 10(b) shows the distribution of the relative measurement error obtained by the proposed method and the ICP. Although the absolute error of the ICP is significantly lower, the relative errors of both algorithms are comparable. It is caused by the fact that the ICP was not able to match the scans after the distance from the initial point increased to more than 2 meters. Therefore, no valid comparison between the cross-correlation and the ICP is available for these longer distances and the relative errors turn out similar. The upper and lower adjacent values of the population of all measured relative errors obtained by the cross-correlation method were −4.23% and 4.92%, respectively.

6. Conclusions and Future Work

A new scan matching algorithm based on cross-correlation is proposed and evaluated in this work. It was devised as an efficient mobile robot localization method with high accuracy and low computational costs. Extensive practical experiments, conducted within the scope of this research, have shown that the proposed algorithm is able to match LiDAR scans with a high accuracy and is robust towards dynamic changes in the environment and moving objects.

The proposed method is computationally lightweight, suitable for the use in low-power mobile devices, and has no special requirements on sensor subsystems. It has been extensively tested and compared with a standard scan matching method, ICP. The comparison has shown that it is approximately 10 times faster and has a wider operating range. The disadvantages of the proposed method include the relatively rough resolution of the obtained transformation
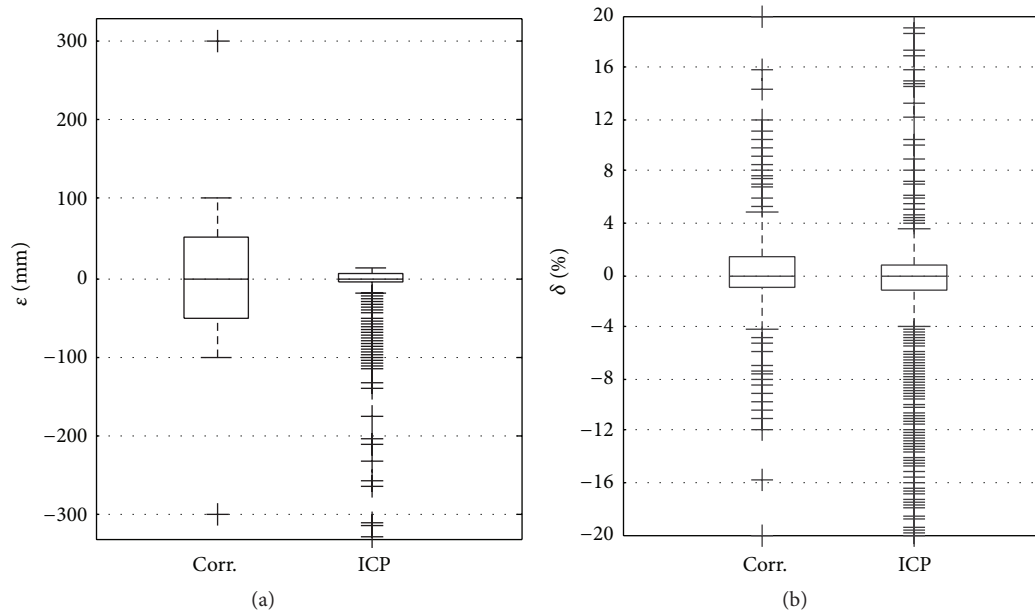
FIGURE 10: Error comparison: (a) absolute error, (b) relative error.

vectors, slightly higher relative error (approx. 5%), and the need to set a fixed resolution of the raster. Raster resolution, suitable for the employed LiDAR sensor and the intended application, was set to 200 mm.

This work can continue in several directions. Advanced methods for real parameter optimization, including, for example, bioinspired optimization metaheuristics [39], can be employed to determine the optimum scan matching parameter vector. The resolution of the proposed method can be improved by seeking an optimum raster cell size with a good trade-off between time complexity and matching accuracy.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

References

[1] D. Filliat and J.-A. Meyer, "Map-based navigation in mobile robots: I. A review of localization strategies," *Cognitive Systems Research*, vol. 4, no. 4, pp. 243–282, 2003.

[2] J. Machaj and P. Brida, "Impact of radio map simulation on positioning in indoor environtment using finger printing algorithms," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 15, pp. 6404–6409, 2015.

[3] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, Bradford Company, Scituate, Mass, USA, 2004.

[4] S. Ge and F. L. Lewis, *Autonomous Mobile Robots*, CRC/Taylor, Boca Raton, Fla, USA, 2006.

[5] Y. Liu and H. Zhang, "Towards improving the efficiency of sequence-based SLAM," in *Proceedings of the 10th IEEE International Conference on Mechatronics and Automation (ICMA '13)*, pp. 1261–1266, Takamatsu, Japan, August 2013.

[6] A. Burguera, G. Oliver, and J. D. Tardos, "Robust scan matching localization using ultrasonic range finders," in *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 1367–1372, August 2005.

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

[8] X. Zezhong, L. Jilin, and X. Zhiyu, "Scan matching based on CLS relationships," in *Proceedings of the IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, vol. 1, pp. 99–104, Changsha, China, October 2003.

[9] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, vol. 3, pp. 2743–2748, IEEE, October 2003.

[10] C. Friedman, I. Chopra, and O. Rand, "Perimeter-based polar scan matching (PB-PSM) for 2D laser odometry," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 80, no. 2, pp. 231–254, 2015.

[11] A. Mallios, P. Ridao, D. Ribas, and E. Hernández, "Scan matching SLAM in underwater environments," *Autonomous Robots*, vol. 36, no. 3, pp. 181–198, 2014.

[12] T. Roferi, "Using histogram correlation to create consistent laser scan maps," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 625–630, Lausanne, Switzerland, 2002.

[13] L. Armesto and J. Tornero, "Slam based on kalman filter for multi-rate fusion of laser and encoder measurements," in *Proceedings of the IEEE/RSJ International Conference on*

Intelligent Robots and Systems (IROS '04), vol. 2, pp. 1860–1865, Sendai, Japan, October 2004.

[14] LMS100, 2015, http://www.sick.com/group/EN/home/products/ product_news/laser_measurement_systems/Pages/lms100.aspx.

[15] H. Ryu and W. K. Chung, "Efficient scan matching method using direction distribution," *Electronics Letters*, vol. 51, no. 9, pp. 686–688, 2015.

[16] Z. Xuehe, L. Ge, L. Gangfeng, Z. Jie, and H. ZhenXiu, "GPU based real-time SLAM of six-legged robot," *Microprocessors and Microsystems*, 2015.

[17] Y. Gao, S. Liu, M. M. Atia, and A. Noureldin, "INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm," *Sensors*, vol. 15, no. 9, pp. 23286–23302, 2015.

[18] S. Bando and S. Yuta, "Use of the parallel and perpendicular characteristics of building shape for indoor map making and positioning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 4318–4323, IEEE, Taipei, Taiwan, 2010.

[19] N. Zikos and V. Petridis, "6-DoF low dimensionality SLAM (l-SLAM)," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 79, no. 1, pp. 55–72, 2015.

[20] J. Li, J. Bao, and Y. Yu, "Localization for a rescue robot based on Ndt scan matching," *Key Engineering Materials*, vol. 439-440, pp. 445–450, 2010.

[21] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling (3DIM '01)*, pp. 1–8, 2001.

[22] L. R. Muñoz and J. J. A. Pimentel, "Robust local localization of a mobile robot using a 180˚2-D laser range finder," in *Proceedings of the 6th Mexican International Conference on Computer Science (ENC '05)*, pp. 248–255, September 2005.

[23] X. Zezhong, L. Jilin, and X. Zhiyu, "Map building and localization using 2D range scanner," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, vol. 2, pp. 848–853, July 2003.

[24] J. Diebel, K. Reutersward, S. Thrun, J. Davis, and R. Gupta, "Simultaneous localization and mapping with active stereo vision," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, vol. 4, pp. 3436–3443, Sendai, Japan, September 2004.

[25] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," *The International Journal of Robotics Research*, vol. 26, no. 10, pp. 1125–1153, 2007.

[26] M. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based SLAM," *International Journal of Robotics Research*, vol. 27, no. 6, pp. 667–691, 2008.

[27] M. Karahan, A. M. Erkmen, and I. Erkmen, "Prioritized mobile robot exploration based on percolation enhanced entropy based fast SLAM," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 75, no. 3-4, pp. 541–567, 2014.

[28] E. Tsardoulias and L. Petrou, "Critical rays scan match SLAM," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 72, no. 3-4, pp. 441–462, 2013.

[29] S. Saeedi, L. Paull, M. Trentini, and H. Li, "Neural network-based multiple robot simultaneous localization and mapping," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, part 2, pp. 2376–2387, 2011.

[30] J. Tang, Y. Chen, X. Niu et al., "LiDAR scan matching aided inertial navigation system in GNSS-denied environments," *Sensors*, vol. 15, no. 7, pp. 16710–16728, 2015.

[31] P. Loncomilla and J. R. Del Solar, "Visual SLAM based on rigid-body 3D landmarks," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 66, no. 1-2, pp. 125–149, 2012.

[32] X. Zhang, A. B. Rad, Y.-K. Wong, Y. Liu, and X. Ren, "Sensor fusion for SLAM based on information theory," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 59, no. 3-4, pp. 241–267, 2010.

[33] S. Lee, S. Lee, and S. Baek, "Vision-based kidnap recovery with SLAM for home cleaning robots," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 67, no. 1, pp. 7–24, 2012.

[34] J. Konecny, M. Kelnar, and M. Prauzek, "Advanced waste rock exploring by mobile robot," *Applied Mechanics and Materials*, vol. 313-314, pp. 913–917, 2013.

[35] A. Lilienthal and T. Duckett, "Building gas concentration gridmaps with a mobile robot," *Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 3–16, 2004.

[36] A. Kurecka, J. Konecny, M. Prauzek, and J. Koziorek, "Monte carlo based wireless node localization," *Elektronika ir Elektrotechnika*, vol. 20, no. 6, pp. 12–16, 2014.

[37] R. P. Singh and S. D. Sapre, *Communication Systems*, Tata McGraw-Hill, New Delhi, India, 2nd edition, 2007.

[38] K. Weiler and P. Atherton, "Hidden surface removal using polygon area sorting," *ACM SIGGRAPH Computer Graphics*, vol. 11, no. 2, pp. 214–222, 1977.

[39] N. Corso and A. Zakhor, "Indoor localization algorithms for an ambulatory human operated 3D mobile mapping system," *Remote Sensing*, vol. 5, no. 12, pp. 6611–6646, 2013.