ELSEVIER

Robotics and Autonomous Systems 31 (2000) 213–225

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

A two-stage mobile robot localization method by overlapping segment-based maps☆

Antonio Reina*, Javier Gonzalez

Dpto. Ingenieria de Sistemas y Automatica, University of Malaga, Plaza El Ejido s/n, 29013 Malaga, Spain

Received 21 September 1998; received in revised form 26 April 1999; accepted 20 August 1999
Communicated by F.C.A. Groen

Abstract

This paper presents a new method for accurately estimating the pose (position and orientation) of a mobile robot by registering a segment-based local map observed from the current robot pose and a global map. The method works in a two-stage procedure. First, the orientation is determined by aligning the local and global map through a voting process based on a generalized Hough transform. Second, it uses a coarse-to-fine approach for selecting candidate positions and a weighted voting scheme to determine the degree of overlap of the two maps at each of these poses. Unlike other methods previously proposed, this approach allows us to uncouple the problem of estimating the robot orientation and the robot position which may be useful for some applications. In addition it can manage environments described by many (possibly short) segments. This paper presents some experimental results based on our mobile robot RAM-2 that show the accuracy and the robustness of the proposed method even for poor quality maps and large dead-reckoning errors.

Keywords: Mobile robot self-localization; Radial laser scanner; Segment-based map; Map registration; Hough transform

1. Introduction

In a two-dimensional space (2D), mobile robot self-localization consists of estimating the translation (t_x, t_y) and orientation θ of the mobile coordinate system (robot frame) with respect to an absolute coordinate system (world frame) (see Fig. 1).

This problem has received great attention in the literature and a variety of techniques have been proposed to address it (for a survey see [21] or [1]). Since methods using trajectory integration (dead-reckoning or inertial navigation systems) suffer from error accumulation, registration with the environment is required. Consequently, the key issue in robot localization is that of properly matching sensor data to a world model.

In this paper we assume that the robot is provided with a 2D segment-based *global map* (referred to a world frame) and a robot-centered *local map* that approximates the shape of the surrounding objects. Both local and global maps are constructed from the data supplied by a radial laser rangefinder.

Our radial laser rangefinder, called the Explorer, is a time-of-flight range scanner, manufactured by Schwartz Electro-optics Inc. The components of the

☆This work has been supported by Spanish Government under project CICYT-TAP96-0763.

*Corresponding author. Fax: +34-5-2131413.
E-mail addresses: reina@ctima.uma.es (A. Reina), jgonzalez@ctima.uma.es (J. Gonzalez)

0921-8890/00/$ – see front matter
PII: S0921-8890(99)00098-6

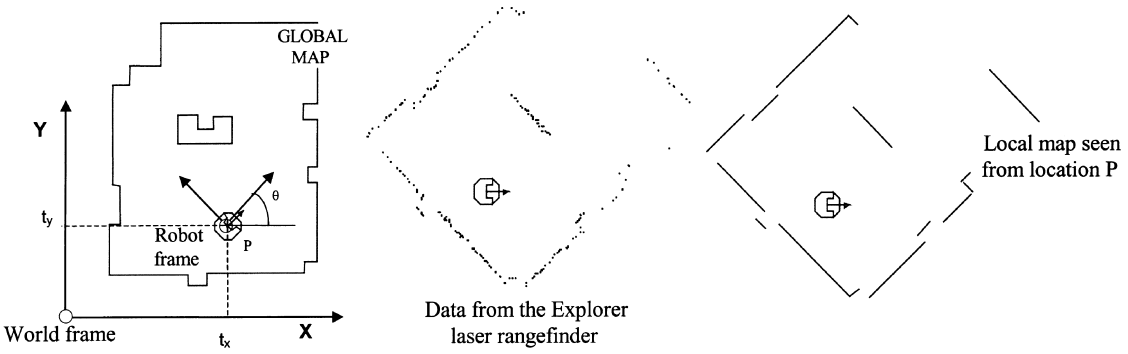

Fig. 1. World frame and robot frame. The global map is referred to the world frame while the local map is expressed in the robot frame.

Explorer are an emitter/receiver pulsed gallium infrared laser, a rotating prism, a driving motor, and an encoder mounted on a steel housing (Fig. 2). By rotating the prism, the Explorer scans 360° field-of-view in a plane parallel to the ground, providing a two-dimensional description of the environment in polar coordinates. The angular resolution can be programmed to measure 128, 256, 1024 and 2048 data per revolution, and the rotation speed is programmable between 0.5 and 4 revolutions per second [17].

Our approach for pose (position and orientation) estimation is basically geometrical, in contrast to other approaches to the problem that are stated from a statistical viewpoint. Because of the characteristics of the laser range measurements (accuracy, repeatability, influence of surface material, etc.) we think that sensor errors are better modeled by a bounded region than by a probabilistic model. For example, based on the calibration of the Explorer, we have checked that, for a given surface material, the range measurements are affected by a gaussian noise with a standard deviation of 1.2 cm. However, other parameters such as sensor resolution, incidence angle and reflectance properties of the surface give rise to much larger errors, which cannot be considered within a statistical framework. In particular, some surface materials (i.e. glazed tiles, polish metal, etc.) can produce errors of more than 15 cm. [17]. On this basis, we see that it is more appropriate to approach the matching problem through a geometrical viewpoint.

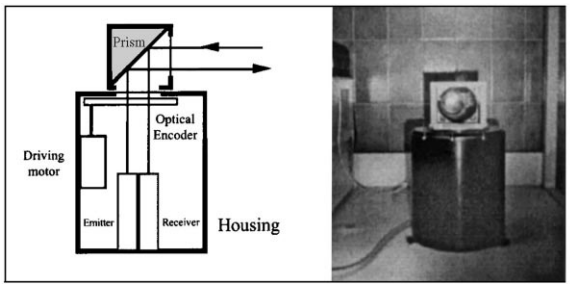

Fig. 2. The Explorer laser radial scanner.

The proposed localization system works in a two-stage procedure:

1. *Map alignment*. Robot orientation is determined through the best alignment of the global and the local maps. This process is accomplished by means of a weighted generalized Hough transform [6]. The key idea consists of rotating the local map through a tentative angle and then checking for the number of equally oriented segments from both the global and local maps, regardless of their relative positions. This algorithm determines the robot orientation without explicitly solving the correspondence between segments. In our tests, the accuracy achieved in the estimated robot orientation is less than 2.5°.
2. *Map overlap*. The position of the robot is estimated by an iterative search process based on a coarse-to-fine scheme. At each iteration, this process generates a set of candidate robot positions that is used to refer the local map to the world frame. For all of these positions the degree of overlap between the local and global maps is computed

by solving the correspondence between segments of both maps. The robot position that produces the maximum overlap is selected as the initial robot position for the next iteration. Obviously, at each iteration of the coarse-to-fine process, the conditions for establishing the correspondence between segments become more restrictive. [1] In order to improve the final pose estimate a refinement of the orientation provided by the first stage is also permitted.

The rest of the paper is organized as follows. We first review some related work. Then the map alignment process is presented. In Section 4 we describe the map overlapping algorithm. In Section 5, the overall method is extensively tested using both synthetic and real maps obtained from our mobile robot RAM-2. Finally, some conclusions and future work are outlined.

2. Related work

As stated above, mobile robot pose estimation turns out to be the problem of matching different observations of the same pieces of the world. We can distinguish two different approaches to match 2D range data: iconic and feature-based methods.

In *iconic methods*, registration is accomplished without explicitly using the underlying features existing in the range scan. One possibility is that of matching the range points against features of a model (probably obtained from previous scans). The position estimators proposed in [4,11,16] are some examples of this scheme.

A second possibility is to match the range scan to previously acquired raw data. In [15], Lu presents an iterative algorithm that establishes correspondences for data points by combining two rules: a *closest-point rule*, that chooses the closest point in the next scan as the correspondence for the data point, and a *matching-range-point rule* that assumes there is no translation between the two scans and chooses for correspondence the closest point with the same range. Our previous work [12] estimates the transformation between consecutive scans along a path based on the spatial and temporal linearization of the range function of each point. Both methods, in particular the latter, are restricted to relative small displacements between the maps.

Of special significance is also the work presented in [23], where the concept of *angle histogram* is used to represent a statistic of the distribution of angles of vector differences (vector that is joined to consecutive points in the scan) with respect to a symmetry axis of the robot system. The orientation is estimated as the phase shift between two histograms computed by a cross-correlation function. Position is determined in a similar way using x–y histograms. Good estimates of the position and orientation of the robot moving along a previously traversed path are reported. However, the method seems to be quite dependent on a good initial estimation as well as on the proximity to the closest available reference histograms. Another disadvantage of this approach is the difficulty in maintaining the maps due to the large amount of data required [22].

Recently, Crowley et al. [5] have presented a mathematically innovative technique that transforms a range scan into a single point in the multidimensional component space. Then, based on principal component analysis, an eigenspace is constructed where the structure of the environment is represented as a family of surfaces. Since the mapping between the eigenspace and the pose space is not unique, a list of candidate pose points (the nearest) is given for a particular range scan. These candidates are tested using different methods in order to select the best one. Unfortunately, this work does not show sufficient experimental results to assess the accuracy and the robustness of the method in more sophisticated clutter environments. In addition, it is an exhaustive method that requires a large amount of memory.

Finally, although more suitable for sonar readings, [2] we must mention in this category the well-known occupancy grid techniques [9,18].

In comparison with feature-based methods, the iconic scheme presents the advantage of not requiring the extraction of distinctive features (usually seg-

[1] Since the data used to build both the global and local maps are not error-free [17], they will not overlap perfectly even for an exact robot pose estimation.

[2] Due to the accuracy of the range measurements, any acceptable approximation of range scan would require a extremely high dense grid, which would result impractical.

ments) from the range scan. Thus, it leads to a less restrictive method, appropriate for matching smooth free-form shapes. In general, its major drawbacks are that it is limited to relatively small displacements between the two scans [19] and it is more sensitive to noise in the range data.

In *feature-based methods*, registration is accomplished by first extracting a set of features from the scan (usually segments and corners) and then making correspondences between pairs of features of the acquired scan and a global map. Then, the robot location is estimated as the rigid transformation that minimizes a certain function that combines the distance errors between the corresponding pairs established. Some examples of this approach are the works reported in [3,8,10,16,18,20]. An interesting approach that, as we do, uses the Hough transform is presented in [10]. Although their algorithm provides excellent results when navigating in indoor cluttered environments, it is restricted to operating in rectangular-shaped scenarios where no more than two predominant walls are present in each direction. Also, some researchers have used the Hough transform to find line segments either in occupancy grids [18] or directly in the range scan [7,10]. The work presented by Dubrawsky and Siemiątkowska [7] is based on a modification of the *angle histogram* method [23] aimed at making the algorithm more robust to measure noise by extracting segments from the scan.

Although the approximation of the range map to features (segments, corners, clusters, etc.) is a time consuming process and leads to an inevitable loss of information, in the context of mobile robotics this turns out to be not so problematic since building and maintaining the map of the environment is highly desirable.

The method proposed in this paper is a feature-based approach and, besides its simplicity and the accuracy it provides, contributes to three major issues:

1. Usually, in the feature-based approach, the discrepancy to minimize refers to the supporting lines of the segments (i.e. perpendicular distance, angle, etc.) [10,16,20]. Contrary to this, we explicitly deal with the line segments. This makes the method more accurate while allowing us to cope with more complicated environment suitable to be approximated by short line segments. To guarantee robustness to noisy points, the segments must come up from a reasonable number of points [13]. In addition, as it will be explained later, the contribution of each segment to the estimated pose must be proportional to its length.
2. As the methods proposed by Weiβ and Wetzler [23] and Dubrawski and Siemiątkowska [7], our approach uncouples the problem of estimating robot orientation and robot position. We believe that precise knowledge of the robot heading by itself has great importance for two main reasons. First, the estimation of the robot position now becomes a much easier problem. Second, in some situations only the orientation is relevant for the navigation task. For instance, when a robot is moving through a corridor the position of the robot is irrelevant while the heading has to be kept within a narrow interval. Another kind of situation occurs when the navigation task is commanded in terms of steering information, for example: “moves straight on until the first junction appears”.
3. No initial robot orientation is required and only a rough initial position (depending on the particular environment) is necessary. Similar capabilities are achieved by the algorithms of [2,5].

In the following sections we discuss in detail the two stages of our method.

3. Map alignment

The idea of this stage consists of aligning the local map (LM) and the global map (GM) through a weighted generalized Hough transform. The Hough transform is a technique frequently used to detect curves of a given shape in an image. This classical Hough transform requires that the curve be specified in some parametric form and, hence, is most commonly used in the detection of regular curves as lines, circles and ellipses, etc. On the other hand, the generalized Hough transform is capable of detecting arbitrary curved shapes by first constructing a reference table using a prototype shape [6].

The algorithm proposed here works according to the following three steps:

1. *Building the reference table* (*R-table*). The R-table is a look-up table that stores the length λ_{GS} of each segment of the global map indexed by its absolute orientation β_{GS}. The orientation β_{GS} represents the

slope of the line supporting the segment and is quantified at integer values between 0° and 180°. The length λ_{GS} is expressed within a resolution of millimeters. In order to account for more than one segment of the global map having the same orientation, the R-table has to record as many lengths λ_{GS} as segments exist for each particular orientation β_{GS}, i.e.

$$\text{R-table}(\beta_{GS}) = [\lambda_{GS}(1), \lambda_{GS}(2), \dots].$$

If the GM is fixed this table will be computed just once. However, each time the GM is updated, for example by updating it with a new LM, the R-table needs to be reconstructed. In any case, this process does not require too much computation time.

2. *Defining the accumulation array*. The accumulation array is a vector **AC** indexed by all the possible discrete robot orientations α (in degrees) with respect to the world frame. Each element $\mathbf{AC}(\alpha)$ represents the number of votes that the orientation α receives according to the voting process described below. The orientation that receives the greatest number of votes will correspond to the best alignment between LM and GM. The range of possible robot orientations depends on the prior information that the system may have. If no information is available, the accumulation array **AC** will consider the full orientation range, i.e. from 0° to 360°. In the case of an initial estimation α_0 and a bounded uncertainty interval δ being available, for instance provided by dead-reckoning, the array **AC** would account only for the candidates in this interval, i.e. $\alpha \in [\alpha_0 - \delta, \alpha_0 + \delta]$. The array **AC** is initialized to zero.
3. *The voting process*. The idea of the voting process consists of rotating the LM through a tentative angle α and then checking for the number of equally oriented segments from both the global and the local maps, regardless of their relative positions (see Fig. 3).

More precisely, for a particular robot orientation α, the new angle β_{LS} of a segment of an LM is computed by

$$\beta_{LS} = \sigma_{LS} + \alpha,$$

where σ_{LS} is the slope of the segment in the robot frame. Now, we look for the entries of the R-table indexed by β_{LS} and increment $\mathbf{AC}(\alpha)$ according to the following voting scheme:

$$\mathbf{AC}(\alpha) = \mathbf{AC}(\alpha) + \lambda_{LS}\,\lambda_{GS}, \tag{1}$$

where λ_{LS} is the length of the local segment being considered and λ_{GS} the most similar length among the entries indexed by β_{LS}.[3] Obviously, in the event that there is no entry for a particular β_{LS}, the accumulator **AC** is not modified. Notice how this weighting process takes into consideration the intuitive fact that the longer the paired segments are the more certain is the robot orientation.

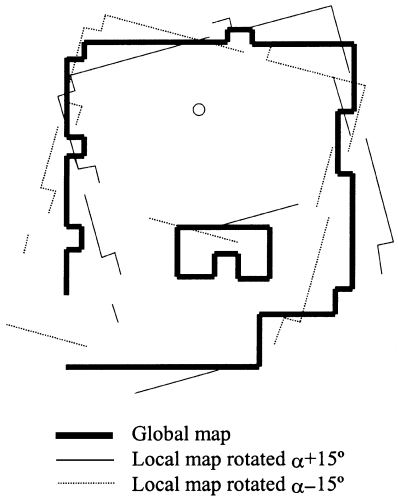

Fig. 3. Two tentative alignments between a local and global maps.

Once all the angles α have been processed, the maximum of the accumulator **AC** provides the estimated robot orientation. In Fig. 4 a pseudo-code summarizes the complete algorithm.

We have also tested other voting strategies, including:

$$\mathbf{AC}(\alpha) = \mathbf{AC}(\alpha) + 1, \tag{2}$$

$$\mathbf{AC}(\alpha) = \mathbf{AC}(\alpha) + \lambda_{SL}, \tag{3}$$

$$\mathbf{AC}(\alpha) = \mathbf{AC}(\alpha) + (\lambda_{SL} + \lambda_{SG}), \tag{4}$$

[3] When no additional information is considered (for example regarding the position of the robot), the global segment whose length λ_{GS} is more similar to λ_{LS} is considered as the best candidate to be voted. Experimentally, we have also checked that other possible voting strategies with more than one global segment involved yield worse results.

1.- Build the **R-table** using the segments in the **GM**
R-table (β_{GS}) = [$\lambda_{GS}(1)$, $\lambda_{GS}(2)$, ...]

2.- Quantize the accumulator array: identify the maximum and minimum values of possible robot orientations:
$\alpha_{maximum} = \alpha_o + \delta$
$\alpha_{minimum} = \alpha_o - \delta$

3.- Generate and initialize the accumulator array (**AC**),
AC (α) = 0; $\alpha = \alpha_{minimum} .. \alpha_{maximum}$

4.- For all the possible robot orientations α do
For each local segment **LS** do
compute slope σ_{LS} and length λ_{LS} of **LS**
rotate **LS**, $\beta_{LS} = \sigma_{LS} + \alpha$
obtain the entries in **R-Table**(β_{LS})
if only one entry appears then
$AC(a) = AC(\alpha) + \lambda_{LS} \cdot \lambda_{GS}$
if several entries appear
select the λ_{GS} that minimizes $|\lambda_{LS} - \lambda_{GS}|$
$AC(a) = AC(\alpha) + \lambda_{LS} \cdot \lambda_{GS}$

5.- Search for the maximum value in **AC**; the corresponding α gives the estimated robot orientation.

Fig. 4. Pseudo-code.

but they all perform worse than the one proposed above. Note that the simplest voting process with no weighting given by Eq. (2) provides the number of equally oriented segment pairs found in the two maps. A comparison between them is presented in Section 4.

4. Map overlap

Provided that the two maps have been precisely aligned, this stage first uses a coarse-to-fine approach for selecting candidate positions and then determines the degree of overlap of the maps at each of these positions by means of a weighted voting scheme.

4.1. Coarse-to-fine generation of tentative robot positions

Each iteration of the coarse-to-fine process includes the following four steps:

1. Centered at the current estimated robot position (initially the odometric one) a grid of nine points is considered. Each point in this grid represents a candidate robot position. The horizontal and vertical distance between the points of this grid is given by a variable η, whose initial value depends on the certainty of the odometric robot position.
2. At every point of the grid three different robot orientations are also considered as candidates[4] (see Fig. 5). These orientations are $\alpha + \Delta\alpha$, α and $\alpha - \Delta\alpha$, where α is the current estimated robot orientation (initially the orientation supplied by the maps alignment process) and $\Delta\alpha$ is an angular increment whose initial value depends on the certainty about the initial robot orientation.
3. At every candidate robot pose considered in steps 1 and 2 a measure of the overlap between the local and the global map is calculated. Broadly, the degree of overlap of the two maps is computed as the sum of the length of the segments that are in correspondence. This process will be described in detail in Section 4.2.
4. A new estimated pose is obtained by the weighted average of those grid poses whose degree of overlap exceeds a given threshold. The threshold is a specific ratio of the maximum overlap computed (for example 90%). If it is the last iteration the algorithm ends, otherwise a new iteration starts using:

$$\eta = \tfrac{1}{2}\eta, \quad \Delta\alpha = \tfrac{1}{2}\Delta\alpha.$$

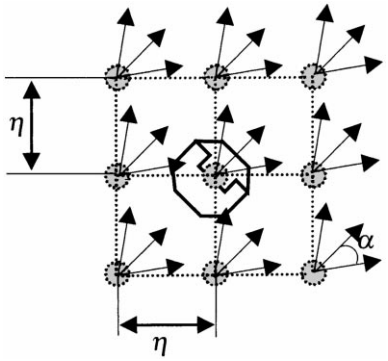

Fig. 5. Grid of candidate robot poses considered at each iteration of the coarse-to-fine process.

[4] Although the robot orientation has been previously computed, the consideration of small variations in the robot orientation enables improving the final estimated orientation.

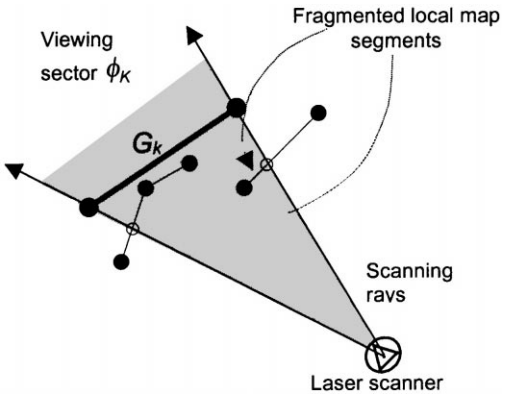

Fig. 6. Viewing sector of the segment G_k. Fragmentation of the LMS partially included in ϕ_k.

The number of iterations K in the coarse-to-fine process is set in advance and depends on the uncertainty of the initial estimation and on the final accuracy desired. Although the accuracy of the algorithm can be adjusted by modifying the value of K, in practice, it is limited by the quality of the maps that depends on the performance of both the laser scanner and the map builder. Obviously the computational cost of the algorithm grows linearly with K.

4.2. Measuring the overlap between the local and global maps

In order to compute the degree of overlap between the local and global maps it is necessary to check the relationship between segments of both maps. This process is accomplished by analyzing the global map segments (GMSs) one by one by means of its *viewing sector*. [5]

Given a line segment G_k of the global map, the viewing sector ϕ_k is defined as the region subtended by the scanning rays [6] of its endpoints (Fig. 6).

Only those local map segments (LMSs) inside ϕ_k are considered in correspondence with G_k and, more precisely, only the piece of segment within ϕ_k. Therefore, these LMSs partially included in ϕ_k need to be fragmented to isolate the *sub-segments* inside (see Fig. 6).

Once a set of LMS is selected for a particular GMS G_k, it is first necessary to determine which of these segments really correspond to G_k and then compute their degree of overlap. For establishing such a correspondence for a particular LMS L_j the following steps apply (see Fig. 7):

1. The supporting lines of L_j and G_k are computed (line_L, line_G).
2. The two points on line_L, p_1 and p_2, that are at Euclidean distances δ and $-\delta$ from line_G, respectively, are computed. The value of δ is set to be the maximum distance error associated with the selected grid, i.e.

$$\delta = \sqrt{2} \cdot \frac{\eta}{2},$$

   where η is the grid size for this iteration.
3. The part of L_j between p_1 and p_2 (denoted by C_{ls}), if it exists, is selected as the sub-segment corresponding to G_k. Otherwise, if no part of L_j falls between these two points, no correspondence between L_j and G_k is established, and the process ends for L_j.
4. Provided that C_{ls} exists, its corresponding part of G_k (denoted by C_{gs}) is obtained by intersecting its supporting line line_G with the scanning rays of the endpoints of C_{ls}. The sub-segments C_{ls} and C_{gs}

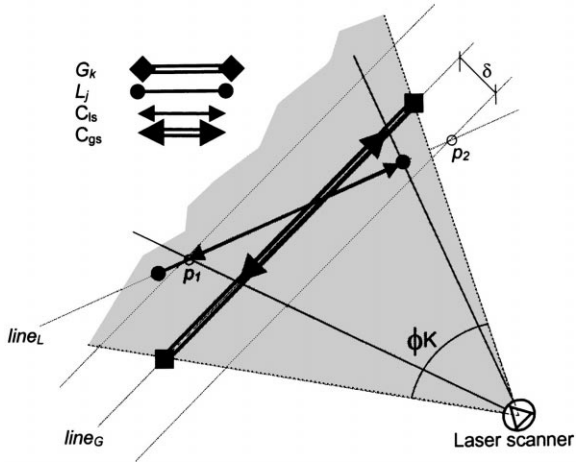

Fig. 7. Solving the correspondence between a global map segment G_k and a local map segment L_j.

[5] This concept has been first introduced by Gonzalez et al. [13], for merging maps.

[6] A scanning ray is a line that passes through the center of the scanner unit.

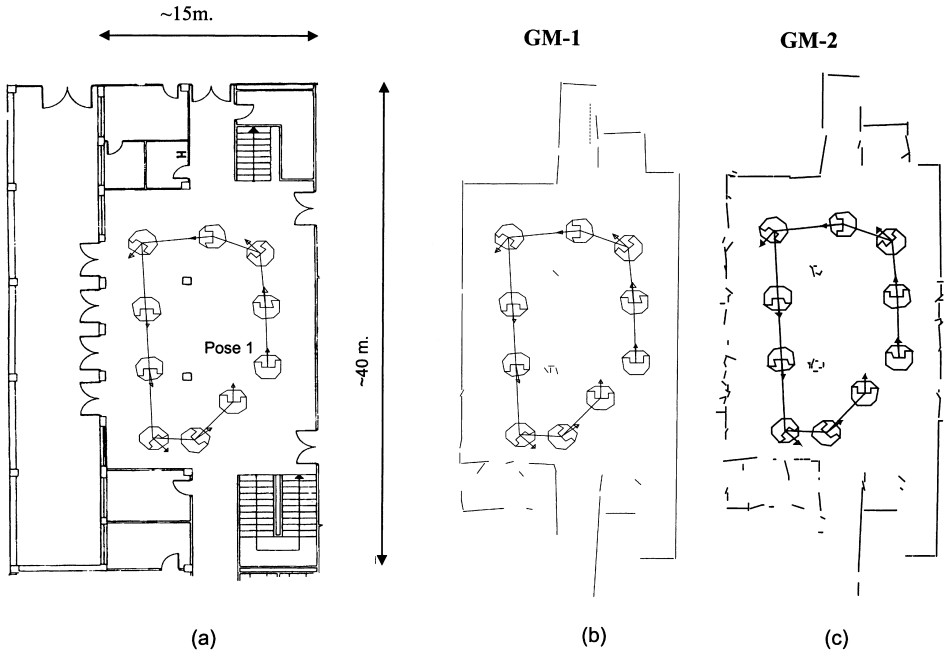

Fig. 8. (a) An architectural plan of the scenario where the tests have been carried out. (b, c) Two global maps of the hall produced by the map builder reported in [14] using two different sets of parameters. GM-1 has 54 segments while GM-2 has 133 segments.

represent the partial correspondence between G_k and LMS.

5. The degree of overlap for the correspondence between G_k and L_j is set to be the sum of the lengths of C_{ls} and C_{gs}.

Once this process has been accomplished for all the segments of the global map, the overall overlap value between the local and global maps is computed as

$$\mathrm{Overlap(GM, LM)} = \sum_{i=1}^{n} \left(\mathrm{length}(C_{\mathrm{gs}}(i)) + \mathrm{length}(C_{\mathrm{ls}}(i))\right),$$

where n is the number of corresponding pairs established.

5. Experimental results

The proposed algorithm has been tested using both synthetic and real data. In the first case, the range measurements used to build the maps are simulated from a probabilistic sensor model that considers the sensor readings to be affected by gaussian noise as well as truncated by a certain resolution [17]. We have used different synthetic environments and in all the cases the errors have remained under 1 cm for robot position and below 0.2° for robot orientation.

The real experiments were conducted using the two global maps shown in Fig. 8. These maps were obtained in the hall near our Mobile Robot Laboratory (Fig. 8(a)), by using the map builder presented in [14] with two different sets of parameters. The main differences between the two maps are the number and size of the segments they contain. The first map, denoted as GM-1, is mostly composed of long segments while the second, denoted as GM-2, exhibits denser clusters of short segments. Note that the orientation of a segment is more reliable the longer the segment.

In this scenario, our RAM-2 mobile robot (see Fig. 9) equipped with the Explorer laser rangefinder was initially positioned at pose 1, and instructed to move to the following locations along the path shown in Fig. 8. At each of these poses the actual robot position and orientation were carefully surveyed by

Fig. 9. The Explorer laser range scanner mounted on the RAM-2 mobile robot.

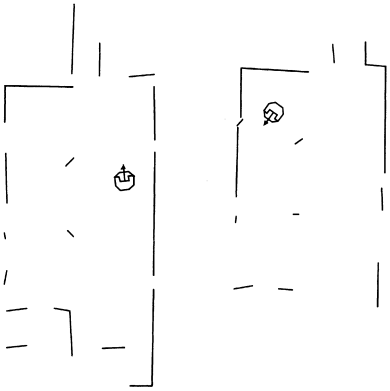

Fig. 10. Local maps obtained from poses 2 and 5 of the mobile robot path.

triangulation using a tape measure and the proposed two-stage algorithm was run. Fig. 10 shows the local maps computed from the second and fifth robot poses in the path (for the sake of clarity these local maps appear aligned to the global frame).

5.1. Testing the map alignment

Since the robot displacements along the path were small, the errors in the robot poses provided by the odometric system were also small. Thus, in order to check the performance of the algorithm for different initial positions and orientations, uniform random values were added to the odometric readings.

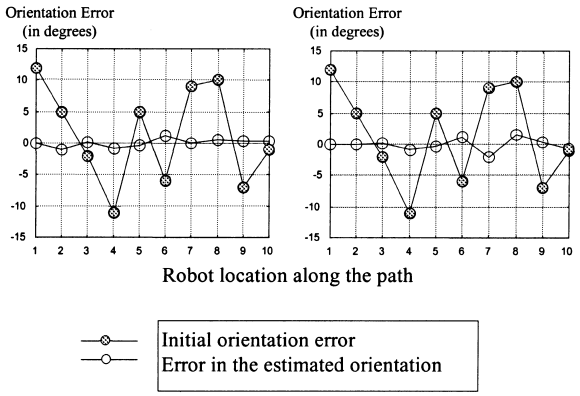

Fig. 11. Orientation errors for the 10 locations along the path when using the GM-1 (left) and GM-2 (right).

In particular, for testing the map alignment stage, at each location along the path the initial robot heading α_0 was taken as the real orientation α_R plus a uniform random error bounded by $\pm 20^\circ$. Thus, the array **AC** was setup to check for all the orientations $\alpha \in [\alpha_0 - 20, \alpha_0 + 20]$.

Fig. 11 plots the errors, calculated as surveyed minus computed, for the 10 locations and for the two global maps. As expected, the errors are slightly greater when using the GM-2 due to the lower quality of their segments. The mean squared errors were 0.3412° and 0.9154°, respectively.

It should be pointed out that the local maps used in each of the two experiments were obtained with the same set of parameters that were used to build the two global maps (shown in Fig. 12). Notice the large discrepancies between the local and the global maps.

To illustrate the performance of the voting process, Fig. 13 shows the number of votes in the array **AC** at location 5 of the path when using the two global maps. Observe that simple counting of the number of pairs found (given by a voting process with no weighting, Eq. (2)) does not produce a clear maximum in the accumulator **AC**. In contrast, the proposed voting scheme, given by Eq. (1), ends up with a clear peak at 1° away from the orientation corresponding to the real heading of the robot.

Finally, we also show the orientation errors provided by the three alternative voting schemes mentioned in

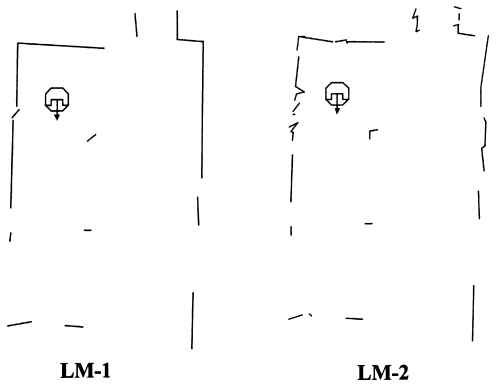

Fig. 12. Local maps obtained at location 5 of the path for two different sets of parameters. LM-1 contains 16 segments while LM-2 has 43 segments.

Section 3. All of these performed worse than the one proposed, although the one corresponding to Eq. (4) gives very similar results which reveals the importance of taking into account the lengths of both matched segments.

5.2. Testing the map overlap

Table 1 shows the errors in the position and orientation (calculated as surveyed minus computed) for the 10 poses along the path. The errors in the robot orientation estimated by the map alignment algorithm also appear in the central column of Table 1, denoted as AE. Notice that these errors, although very small, are significantly reduced after the map overlap procedure. The initial values selected for δ and α in the map overlap process were 1.5 m and 4°, respectively.

It is important to note that these results have been obtained in a very difficult real scenario where stairs, glass doors, small circular columns, people walking around, etc. were present.

The convergence of the algorithm has been extensively tested for different initial positions and orientations at each of the poses of the robot along the path. We have obtained results similar to the one shown in Table 1, except for some infrequent special situations where the algorithm converges to a wrong (although plausible) solution. These situations arise because of the existence of symmetries in the environment along with a very poor initial position. As an example, Figs. 14 and 15 illustrate the track of six points from the initial position up to the final estimated position for five different initial guesses at the seventh pose of the path. The size of the cells of the initial grid was $3.2 \times 3.2\,\mathrm{m}^2$, while the cell size of the final grid was $0.1 \times 0.1\,\mathrm{m}^2$ (which gives us six iterations of the map overlap algorithm with grid sizes: 3.2, 1.6, 0.8, 0.4, 0.2, 0.1). For clarity purpose, the sequence of points has been connected by a spline.

The computational cost of the algorithm is $O(M \times N)$ for the map alignment stage and $O(M \times N \times K)$ for the map overlap stage, where M and N are the number of segments in the global and local maps, respectively, and K is the number of iterations in the coarse-to-fine approach. For the above experiments,

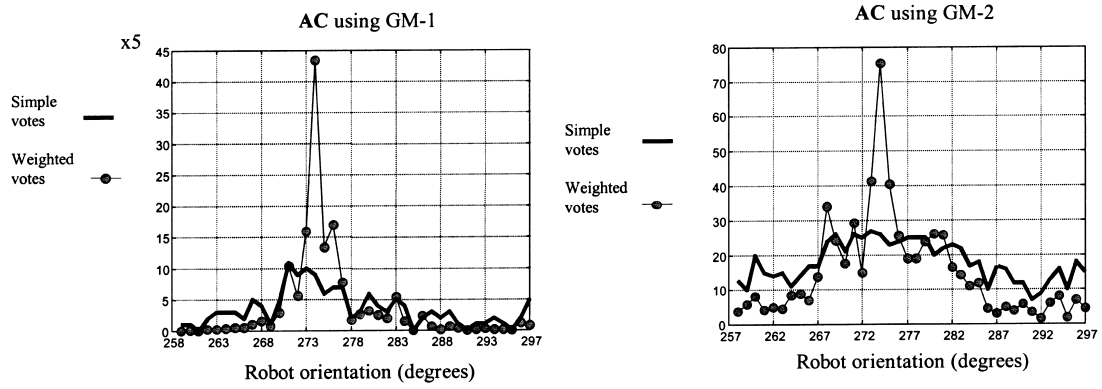

Fig. 13. The final votes in array **AC** at location 5 of the path for both experiments using GM-1 and GM-2. The weighted votes refer to the result of the proposed voting scheme. The simple votes indicate the number of votes with no weighting.

Table 1
Initial and final errors in the robot position estimation[a]

S. No.	Initial errors			AE, α (°)	Estimation errors		
	X (cm)	Y (cm)	θ (°)		X (cm)	Y (cm)	TH (°)
1	182.2	36.0	−9.75	−1.5	2.2	−1.0	−0.07
2	−63.7	49.5	−15.6	0.5	3.7	−2.1	−0.25
3	−91.2	168.5	7.82	2.1	1.2	−1.4	0.18
4	33.2	−131.8	−17.3	−0.3	−3.2	1.5	0.22
5	−10.9	−14.7	−19.9	0.8	0.9	−0.2	−0.24
6	47.2	−168.2	14.1	0.6	2.7	5.1	0.09
7	16.2	38.7	−6.71	−1.3	3.7	−1.2	0.11
8	74.4	−75.1	16.4	−2.3	5.5	−4.8	0.22
9	−56.3	18.1	3.7	0.9	1.3	3.1	−0.09
10	−46.5	−133.6	0.5	1.1	−1.4	−2.3	−0.19

[a] The alignment error (AE) column shows the error in the robot orientation provided by the map alignment procedure.

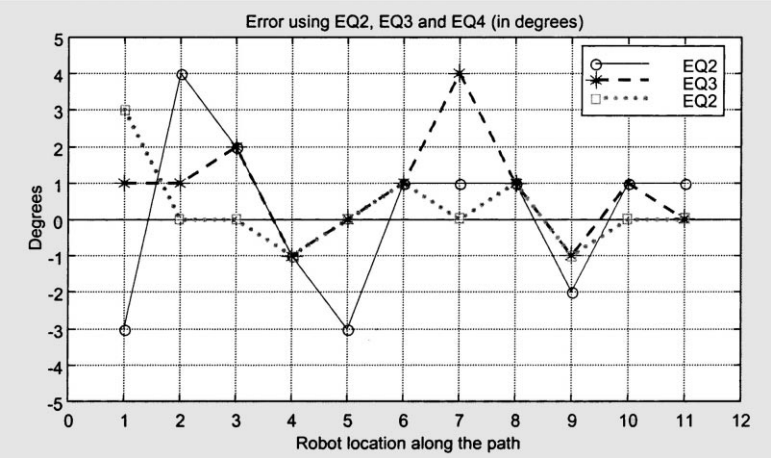

Fig. 14. Orientation errors obtained for the GM-1 using other voting schemes.

with a global map of 52 segments, a number of local map segments ranging from 19 to 31 segments and K equal to 6, the runtimes were always below 1 s on a Pentium 120 MHz under the Lynx Real Time OS.

6. Conclusions and future work

In this paper, a new method for estimating the position of a mobile robot based on registering 2D segment maps has been presented. The method consists of a two-step procedure that first determines the robot orientation and then computes the robot position in addition to refining the orientation.

The first stage, which we have called "map alignment", aligns the two segment maps, regardless of their relative positions, by means of a generalized Hough transform that accounts for the number of equally oriented segments from both the global and local maps. This process does not require any estimation about initial robot orientation and can be useful for a variety of robotic applications where the position estimation is not relevant.

Assuming that an estimated orientation has been computed by this procedure, the second stage uses a grid-based coarse-to-fine approach to derive the selection of candidate poses. At each of these poses the degree of overlap between the observed local map and

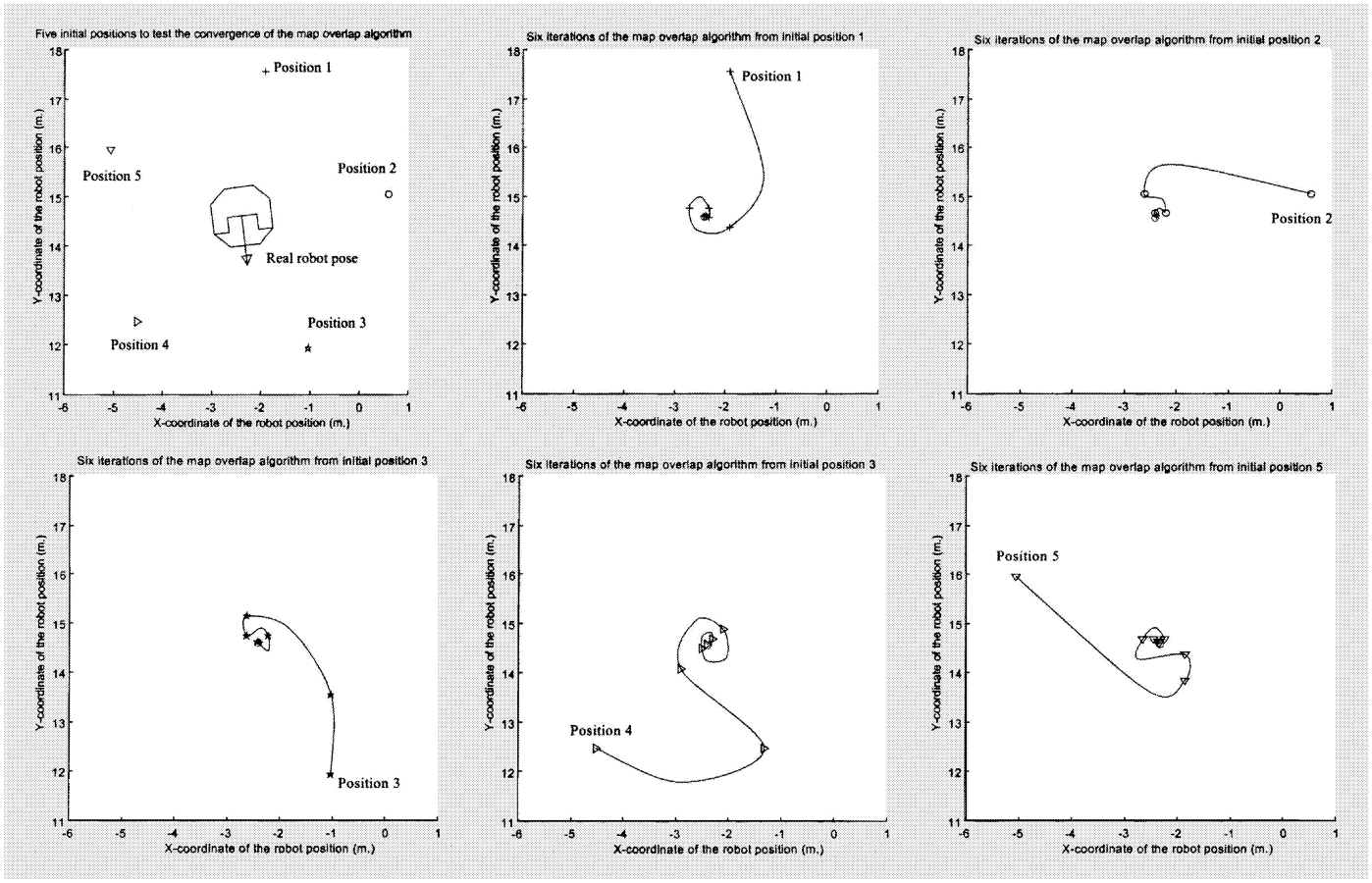

Fig. 15. Sequence of convergence points for five different initial positions at the seventh pose of the robot in the path. The initial size of the cells was $3.2 \times 3.2\,\mathrm{m}^2$ and six iterations of the map overlap process were executed, given a final cell of $0.1 \times 0.1\,\mathrm{m}^2$. For clarity purpose, the sequence of points have been connected by a spline.

a known global map is measured by means of a correspondence process, and a new estimated pose is obtained by the weighted average of those grid poses whose degree of overlap exceeds a given threshold.

This algorithm has been extensively tested using synthetic and real data provided by the Explorer laser range scanner. In particular, we have presented experimental results on our mobile robot RAM-2 that shows the accuracy and robustness of this method even for poor quality maps and large errors in the initial robot position and orientation.

Future work is concerned with the use of a new weighting method that takes into account the degree of confidence in the segments observed. Currently we are developing a map building algorithm that uses the correspondences found in order to merge local and global maps.

References

[1] J. Borenstein, H.R. Everett, L. Feng, D. Wehe, Mobile robot positioning: Sensors and techniques, Journal of Robotic Systems 14 (4) (1997) 213–249.

[2] W. Bugard, A. Derr, D. Fox, A.B. Cremers, Integrating global position estimation and position tracking for mobile robots: The dynamic Markov localization approach, in: Proceedings of the International Conference on Intelligent Robots and Systems, Victoria, BC, October 1998, pp. 730–735.

[3] J.A. Castellanos, J.D. Tardos, Laser-based segmentation and localization for a mobile robot, in: Proceedings of the Second World Automation Congress, Montpellier, France, 1996.

[4] I.J. Cox, Blanche — An experiment in guidance and navigation of an autonomous robot vehicle, IEEE Transactions on Robotics and Automation 7 (2) (1991).

[5] J.L. Crowley, F. Wallner, B. Schiele, Position estimation using principal components of range data, in: Proceedings of the International Conference on Robotics and Automation, Leuven, Belgium, May 1998.

[6] E.R. Davis, Machine Vision. Theory, Algorithms, Practicalities, Academic Press, New York, 1990.

[7] A. Dubrawski, B. Siemiątkowska, A method for tracking pose of a mobile robot equipped with a scanning laser range finder, in: Proceedings of the International Conference on Robotics and Automation, Leuven, Belgium, May 1998.

[8] T. Einsele, Real-time self-location in unknown indoor environments using a Panorama laser range finder, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Grenoble, France, September 1997, pp. 697–702.

[9] A. Elfes, Using occupancy grids for mobile robot perception and navigation, IEEE Computer (June 1989) 46–57.

[10] J. Forsberg, U. Larsson, A. Wernersson, On mobile robot navigation in cluttered rooms using the range weighted Hough transform, IEEE Robotics and Automation Society Magazine (March 1995) 18–26.

[11] J. Gonzalez, A. Stentz, A. Ollero, A mobile robot iconic position estimator using a radial laser scanner, Journal of Intelligent and Robotic Systems 13 (1995) 161–179.

[12] J. Gonzalez, R. Gutierrez, Direct motion estimation from a range scan sequence., Journal of Robotic Systems 16 (2) (1999) 73–80.

[13] J. Gonzalez, A. Ollero, P. Hurtado, Local map building for mobile robot autonomous navigation using a 2D laser range sensor, Proceedings of the IFAC World Congress, Sydney, 1993.

[14] J. Gonzalez, A. Ollero, A. Reina, Map building for a mobile robot equipped with a laser range scanner, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, May 1994.

[15] F. Lu, Shape registration using optimization for mobile robot navigation, Ph.D. Thesis, University of Toronto, Toronto, 1995.

[16] P. MacKenzie, G. Dudek, Precise positioning using model-based maps, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, May 1994.

[17] A. Reina, J. Gonzalez, Characterization of a radial laser scanner for mobile robot navigation, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1997, Grenoble, France, pp. 579–585.

[18] B. Schiele, J.L. Crowley, A comparison of position estimation techniques using occupancy grids, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, May 1994.

[19] G. Shaffer, Two-dimensional mapping of expansive unknown areas, Ph.D. Dissertation, Carnegie Mellon University, Pittburgh, PA, 1995.

[20] G. Shaffer, J. Gonzalez, A. Stentz, A comparison of two range-based pose estimators for a mobile robot, in: Proceedings of the SPIE Conference on Mobile Robots, Boston, MA, November 1992.

[21] R. Talluri, J.K. Aggarwal, Position estimation techniques for an autonomous mobile robot: A review, in: C.H. Chen, L.F. Pau, P.S. Wang (Eds.), Handbook of Pattern Recognition and Computer Vision, World Scientific, Singapore, 1993, pp. 769–801.

[22] G. Weiβ, E.V. Puttkamer, A map based on laser scans without geometric interpretation, in: Proceedings of the 4th Conference on Intelligent Autonomous Systems, Karlsruhe, Germany, March 1995, pp. 172–180.

[23] G. Weiβ, C. Wetzler, E. von Puttkamer, Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans, in: Proceedings of the International Conference on Intelligent Robots and Systems, München, Germany, September 1994, pp. 595–601.