

Article

Curvature-Based Environment Description for Robot Navigation Using Laser Range Sensors

Ricardo Vázquez-Martín ¹, Pedro Núñez ^{2,*}, Antonio Bandera ¹ and Francisco Sandoval ¹

¹ Departamento de Tecnología Electrónica, University of Málaga, E.T.S.I. Telecomunicación, Campus Teatinos, Málaga, Spain; E-Mails: rvmartin@uma.es; ajbandera@uma.es; fsandoval@uma.es

² Departamento de los Computadores y las Comunicaciones, University of Extremadura, Escuela Politécnica, Cáceres, Spain

* Author to whom correspondence should be addressed; E-Mail: pmnt@uma.es; Tel.: + 0034 952 13 1057; Fax: + 0034 952 13 4151

Received: 15 May 2009; in revised form: 22 June 2009 / Accepted: 24 June 2009 /

Published: 24 July 2009

Abstract: This work proposes a new feature detection and description approach for mobile robot navigation using 2D laser range sensors. The whole process consists of two main modules: a sensor data segmentation module and a feature detection and characterization module. The segmentation module is divided in two consecutive stages: First, the segmentation stage divides the laser scan into clusters of consecutive range readings using a distance-based criterion. Then, the second stage estimates the curvature function associated to each cluster and uses it to split it into a set of straight-line and curve segments. The curvature is calculated using a triangle-area representation where, contrary to previous approaches, the triangle side lengths at each range reading are adapted to the local variations of the laser scan, removing noise without missing relevant points. This representation remains unchanged in translation or rotation, and it is also robust against noise. Thus, it is able to provide the same segmentation results although the scene will be perceived from different viewpoints. Therefore, segmentation results are used to characterize the environment using line and curve segments, real and virtual corners and edges. Real scan data collected from different environments by using different platforms are used in the experiments in order to evaluate the proposed environment description algorithm.

Keywords: laser scan data segmentation; mobile robot navigation; adaptive curvature estimation

1. Introduction

Extracting useful information from the environment has an important effect on the robot navigation process. Simultaneous localization and map building (SLAM), path planning, or even a virtual reconstruction of the scene for supervising the robot navigation are different examples where a detailed description of the environment can usually improve their results. To address this issue, an appropriate representation of the working environment of the mobile robot must be acquired, which is not trivial. Many factors and physical constraints affect the reliability of such representation [1].

One of the first tasks in the navigation system design is to determine the type of sensor required to obtain the desired description in a particular environment. The most appropriate sensor for the application depends on the size of the operation area, the environmental conditions, and the required representation level. Indeed, the most important factor that determines the quality of the representation is this external sensor, and above all, its accuracy. With regard to the mobile robotic tasks, an accurate localization in known or unknown environments is essential for autonomous mobile robot navigation. Pure dead-reckoning methods such as odometry are prone to drift, and an estimate is needed to reduce the growing unbounded errors [2]. In order to provide a precise position estimation, external sensors, like sonar or laser range finder sensors, are extensively used in robotics, especially in indoor environments [3–6]. In these sensors, the accuracy is a function of their specifications and the type of features used to represent the environment. Other kinds of commonly used sensors in robotics are cameras, more specifically, monocular, stereo, or trinocular vision systems [7–12]. In these cases, the accuracy of the sensor is a function of the captured image resolution and the features used in the representation.

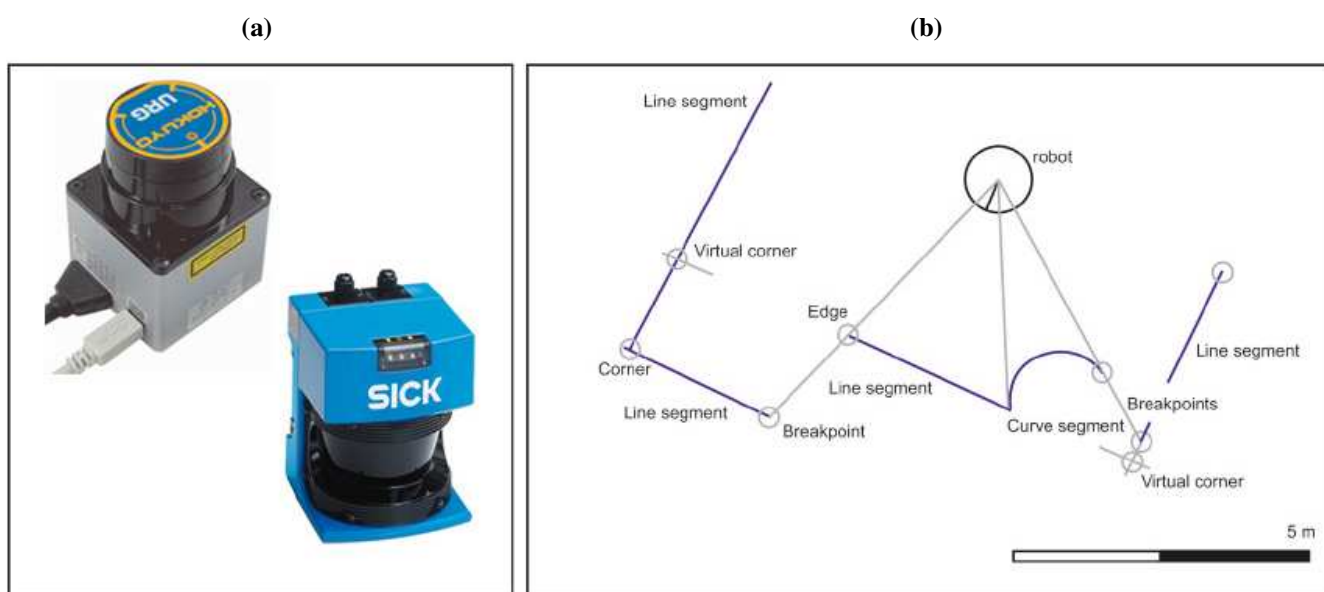
In general, the structural features commonly found in the environment are assumed to be invariant to height (e.g., walls, corners, columns). Using this assumption, a planar representation would be adequate for feature extraction and a distance-based sensor can be used. Among different types of sensors, 2D laser range finders have been increasing popular during the last decade, because they provide dense and accurate range measurements with high angular resolution and sampling rates. Figure 1(a) illustrates two classical laser range sensors used in robotics: a LMS200 from SICK, and a HOKUYO URG-04LX. On the other hand, in terms of cost, it is an affordable device for most mobile robotics systems.

Once the sensor is chosen, the second task that we must address is to match the obtained data with the expected data available in a map. To this end, two approaches have been used in mobile robotics: point-based and feature-based matching. Feature-based approaches increase the efficiency and robustness of this process by transforming the acquired raw sensor data into a set of geometric features. Because they are more compact, these feature-based approaches require much less memory than the point-based approaches and can still provide rich and accurate information [13]. Besides, these methods are more robust to the noise resulted from spurious measurements and unknown objects. Thus, feature-based model is a typical choice for the map representation, which allows the use of multiple models to describe the measurement process for different parts of the environment.

This work extends the CURvature-BAsed (CUBA) approach for environment description: a feature-based approach proposed by Núñez *et al.* [14–16]. In these previous works, the authors present a feature-based approach which employs multiple models to characterize the environment. Specifically, the laser scan is analyzed to detect rupture points, breakpoints and four types of landmarks: line seg-

ments, corners, center of curve segments and edges [14–16] [see Figure 1(b)]. With respect to these previous works, a new laser scan data segmentation based on curvature information is proposed. In order to improve the robustness against noise, this curvature is calculated using a triangle-area representation where the triangle side lengths at each range reading are adapted to the local variations of the laser scan, removing noise without missing relevant points. Besides, in this paper, the proposed environment representation has been used inside a SLAM approach based on the Extended Kalman Filter (EKF).

Figure 1. (a) Two laser range sensors widely used in Robotics: a LMS200 from SICK and a HOKUYO URG-04LX. (b) Natural landmarks detected and characterized in this work: breakpoints, rupture points, line and curve segments, corners and edges.



This work has been organized as follows: Firstly, the most popular methods available in the literature for laser scan data segmentation are briefly described in Section 2. Next, a multi-scale method based on the curvature estimation of the scan data is presented in Section 3. Section 4 describes some improvements to the proposed segmentation module of the approach which have been included in order to increase its robustness against noise and its invariance to translation and rotation. Experimental results and a brief discussion have been included in Sections 5 and 6, respectively. Finally, a brief glossary is given, which includes a list of words related to the robotics field.

2. Laser Scan Data Segmentation Algorithms

2.1. Problem Statement

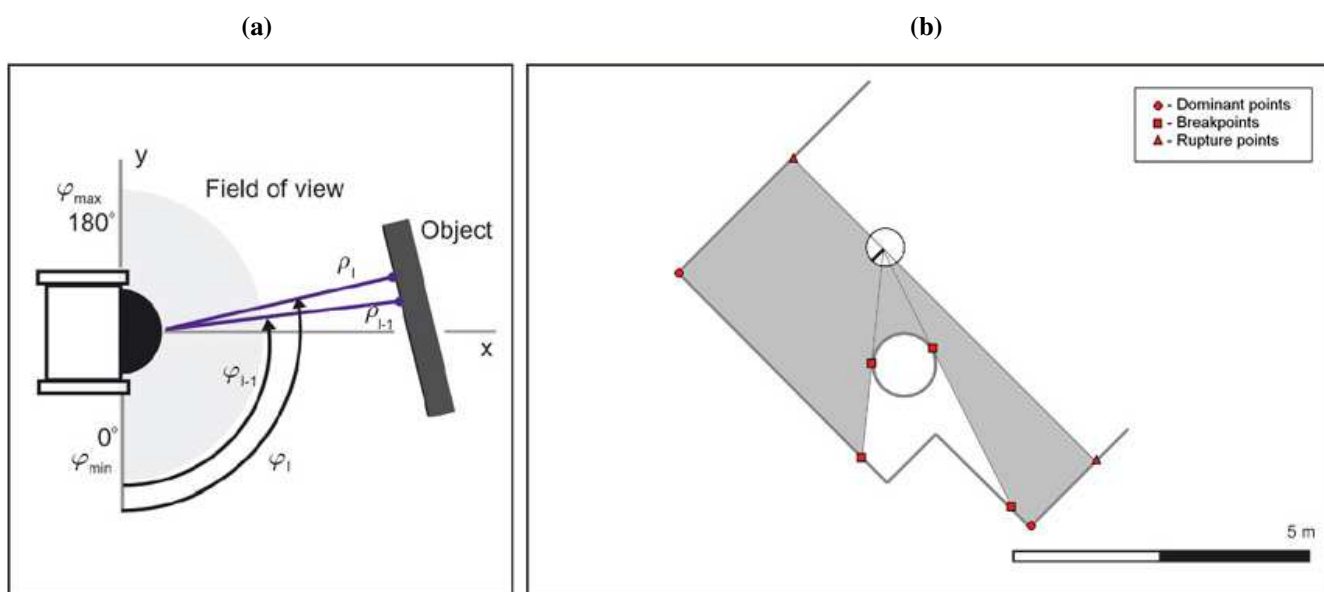
Scan data provided by 2D laser range finders are typically in the form $\{(r, \varphi)_{i=1 \dots N_R}\}$, on which $(r, \varphi)_i$ are the polar coordinates of the i th range reading (r_i is the measured distance of an obstacle to the sensor rotating axis at direction φ_i), and N_R is the number of range readings. Figure 2(a) represents all these variables. It can be assumed that the noise on both measurements, range and bearing, follows a Gaussian distribution with zero mean and variances σ_r^2 and σ_φ^2 , respectively. The aim of segmenting a

laser scan is to divide it into clusters of range readings associated to different surfaces, planar or curves, of the environment. There are two main problems in laser scan segmentation:

- How many segments are there?
- Which range readings belong to which segment?

In order to establish the limits of these segments, these problems can be stated as the search for the range readings associated to the discontinuities in the scanning process or to the changes in the orientation of the scan [see Figure 2(b)].

Figure 2. (a) Scan reference frame variables. (b) Problem statement.



To detect these changes, two main types of techniques have been proposed in the literature. The most popular ones try to find specific geometric features in the scan. Specifically, polygonal approximation techniques originated from computer vision have been widely used to deal with office-like environments, which can be described using line segments. This segmentation process is achieved by checking some heuristic line criteria (i.e., error bound) while concatenating consecutive points. On the other hand, the laser scan data can be represented by a local descriptor which can be analyzed to extract the set of dominant points which correctly segments the scan into curve and line segments.

2.2. Polygonal Based Methods

Among the polygonal-based techniques, the incremental and split-and-merge (SM) approaches are probably the most popular and simple line segments extractors. The split-and-merge algorithm fits a line segment to the set of range readings, and it then divides this line into two new line segments if there is a range reading whose distance to the line is greater than a given threshold. This splitting process is then iteratively applied to the newly generated line segments. Finally, when all line segments have been checked, collinear segments are merged. This algorithm has been used to extract line segments

in many robotic research [17–20]. The incremental algorithm, also known as *Line-Tracking*, starts with two close points and adds the next scan point to the end of the segment when a predefined line condition is satisfied. If the criterion is not achieved, the current line is finished and a new line is started in the next point.

Similar to the SM algorithm, the iterative-end-point-fit method (IEPF) provides a polygonal approximation to the laser scan at a low cost [18]. The procedure is similar to the first part of the SM algorithm. A line is fitted to a set of scan points simply by connecting the end points of two sets. The point with the maximum distance to the line is detected and the set is split up if the distance exceeds a fixed threshold. This splitting process is repeated until the maximum distance is lower than the threshold for all sets.

In order to avoid the need to guess the number of initial clusters, Borges and Aldon [19] employ fuzzy clustering in a split-and-merge framework. The split phase is based on the IEPF method, where for each iteration a set of scan points is divided into two sets if a threshold established for a dispersion measure is not satisfied. Unlike IEPF, the obtained set does not obey the acquired order of the points. In the merge phase, the two closest lines to a reference line are selected as fusion candidates. The fused line is the one that gives the smallest dispersion with the reference line and the given threshold for a single line is fulfilled.

Other model-based popular approaches are based on the Hough transform. The Hough transform has been successfully applied to detect lines in intensity images and it has been brought into robotics for achieving this same aim for scan images [21, 22]. The set of scan data points is sorted into subsets of roughly collinear points using the Hough transform that is based on a voting strategy to determine the best fit for the data subset. The main drawback of this method is the difficulty in choosing a correct size for the voting grid. The parametric space must be discretized and the accuracy is highly affected in real time applications. In order to avoid this problem, the approach proposed by Bandera *et al.* [23] employs a variable bandwidth mean shift algorithm to independently cluster the items of the parameter space in a set of classes.

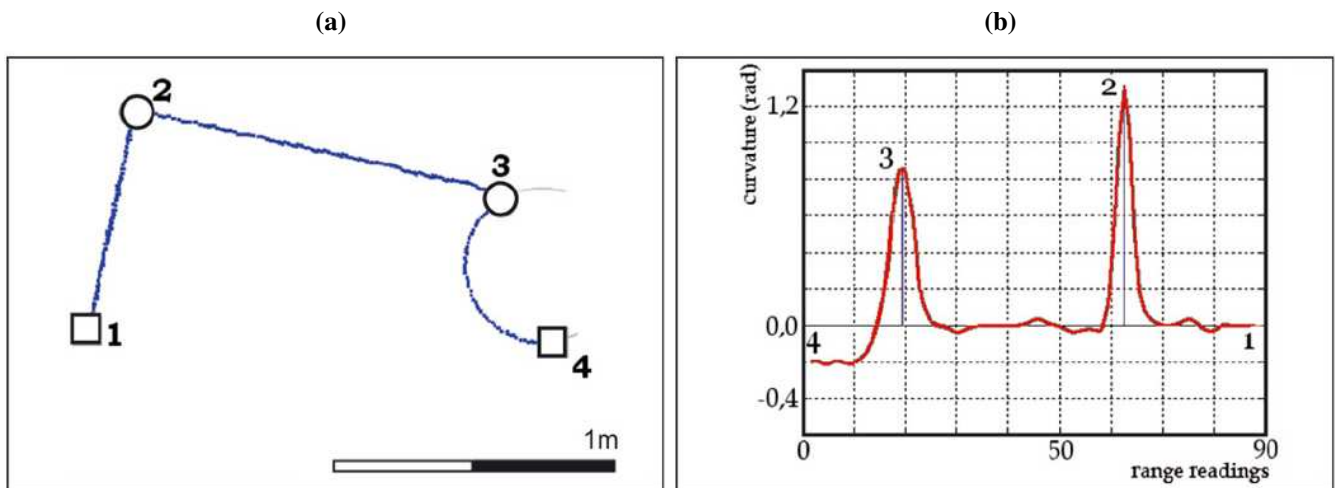
Finally, the aim of *Random Sampling Segmentation Algorithms* is to find a suboptimal probabilistic model to classify the data points and to separate inliers from outliers. Usually, RANdom SAMpling Consensus (RANSAC) is used to detect outliers in a set of data, because it is an efficient algorithm for robust fitting of any kind of models in the presence of data outliers. In this scheme, an algorithm for robust data segmentation is presented in [24]. It is adapted to scale space by using the Adaptive Scale Sample Consensus (ASSC), a modification of RANSAC involving an adaptive scale estimation. The ASSC is a kernel-based scale estimator based on the mean shift method for the data driven scale estimate.

2.3. Curvature-Based Methods

Curvature functions basically describe how much a curve bends at each point. Peaks of the curvature function correspond to the corners of the represented curve and their height depends on the angle at these corners. Flat segments whose average value is larger than zero are related to curve segments and those whose average value is equal to zero are related to straight line segments. Figure 3(a) presents a curve yielding two corners (points 2 and 3) and a curve segment (from point 3 to 4). Peaks corresponding to 2 and 3 can be appreciated in its curvature function [Figure 3(b)]. It also shows that segment 3-4 has an

average value greater than zero, but it is not flat due to noise. Nevertheless, peaks in that segment are too low to be considered as the corners of the curve. Finally, segments 1-2 and 2-3 present a curvature average value near to zero, as expected in line segments.

Figure 3. (a) Segment of a single laser scan (□-breakpoints, o-corners). (b) Curvature function associated to (a).



In a general case, the curvature $\kappa(t)$ of a parametric plane curve, $c(t) = (x(t), y(t))$, can be calculated as [25, 26]

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{3/2}} \quad (1)$$

This equation implies that estimating the curvature involves the first and second order directional derivatives of the plane curve coordinates, (\dot{x}, \dot{y}) and (\ddot{x}, \ddot{y}) , respectively. This is a problem in the case of computational analysis where the plane curve is represented in a digital form [26]. In order to solve this problem, two different approaches have been proposed:

- Interpolation-based curvature estimators. These methods interpolate the plane curve coordinates and then differentiate the interpolation curves. Thus, Mokhtarian *et al.* [25] propose to filter the curve with a one-dimensional Gaussian filter. This filtering removes the plane curve noise.
- Angle-based curvature estimators. These methods propose an alternative curvature measure based on angles between vectors, which are defined as a function of the discrete curve items. Thus, the curve filtering and curvature estimation are mixed by Agam *et al.* [27], which define the curvature at a given point as the difference between the slopes of the curve segments on the right and left side of the point, where slopes are taken from a look-up table. The size of both curve segments is fixed. Liu *et al.* [28] compute the curvature function by estimating the edge gradient at each plane curve point, which is equal to the arctangent of its Sobel difference in a 3x3 neighborhood. Arrebola *et al.* [29] define the curvature at a given point as the correlation of the forward and backward histograms in the k -vicinity of the point, where the resulting value is modified to include concavity and convexity information.

Due to the characteristic noise associated to the curvature estimation, all these algorithms implicitly or explicitly filter the curve descriptor at a fixed cut frequency to remove noise and provide a more robust estimation of the curvature at each plane curve point (*single scale methods*). However, features appear at different natural scales and, since most methods filter the curve descriptor at a fixed cut frequency, only features unaffected by such a filtering process may be detected. Thus, in the case of angle-based curvature estimators, algorithms described above basically consist of comparing segments of k -points at both sides of a given point to estimate its curvature. Therefore, the value of k determines the cut frequency of the curve filtering. In these methods, it is not easy to choose a correct k value: when k is small, the obtained curvature is very noisy and, when k is large, corners which are closer than k points become missing. To avoid this problem, some methods propose iterative feature detection for different cut frequencies, but they are slow and, in any case, they must choose the cut frequencies for each iteration [30]. Another solution is to adapt the cut frequency of the filter at each curve point as a function of the local properties of the shape around it [31].

Both approaches have been applied to laser scan segmentation. Thus, the iterative curvature scale space (CSS) was used by Madhavan and Durrant-Whyte [32] to extract stable corners. This algorithm convolves the curve descriptor with a Gaussian kernel and imparts smoothing at different levels of scale (the scale being proportional to the width of the kernel). From the resulting curve descriptor, features associated to the original shape can be identified [25]. In order to achieve a robust determination of dominant points, the algorithm detects them at the most coarse scale σ_{max} , but localizes the dominant point position at the finest scale σ_{min} . In order to avoid a slow iterative estimation of the curvature, an adaptive algorithm was employed by Núñez *et al.* [16] to extract corners, line and curve segments from the laser scan data.

3. CURvature-BASed Environment Description Framework

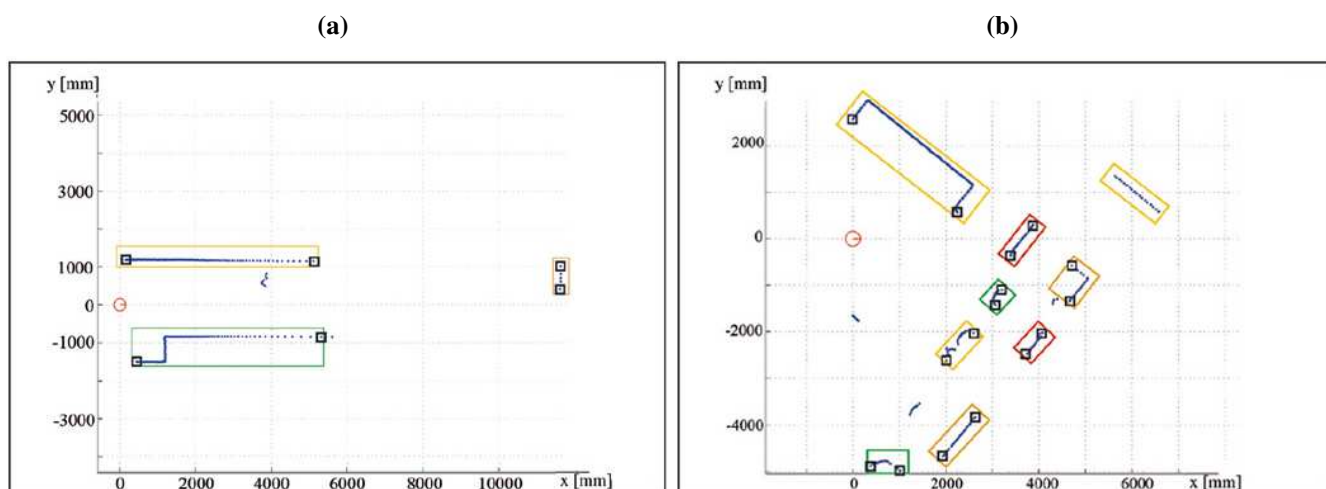
The environment description algorithm described in this paper is divided in two main stages. The first one is a segmentation stage, which divides the scan data acquired by the laser range sensor to a set of point clusters associated to line or corner segments. The next stage detects and characterizes natural landmarks (i.e. line and curve segments, corners and edges) according to these clusters, through both extracting their pose and estimating the uncertainties. This complete characterization allows the use of these features on a later robotic navigation tasks (e.g., SLAM). This work is based on previous papers [14, 16]. In order to improve the robustness against noise in the segmentation stage, this paper includes a novel technique for affine invariant curvature estimation. The new segmentation module is described in the Section 4.

3.1. Segmentation Algorithm

In this section, the segmentation algorithm developed inside the CURvature BASed environment description framework (CUBA in short) is presented. Instead of using a slow, iterative approach, dominant points can be robustly detected by adapting the scale to the local surroundings of each range reading. This solution has been adopted by Núñez *et al.* [14, 16]. The adaptive curvature approach allows to rapidly segment the laser scan into curve and line segments.

In this approach, the segmentation is achieved in two consecutive steps. The first type of segmenting points may arise from the absence of obstacles in the scanning direction (*rupture points*) or from the change of surface being scanned by the sensor (*breakpoints*) [19]. Rupture points cannot be detected by making inferences about its possible presence. They indicate a discontinuity during the measurement and its presence must be informed by the range finder [14]. On the contrary, breakpoints are detected by making inferences about the possible presence of discontinuities in a sequence of valid range data [14, 19]. Basically, the aim of a breakpoint detector is to verify if there exists a discontinuity between two consecutive range readings $(r, \varphi)_n$ and $(r, \varphi)_{n-1}$. This algorithm allows to reject isolated range readings, but it leads to an under-segmentation of the laser scan, i.e., extracted segments between breakpoints typically group two or more different structures (see Figure 4). In order to avoid this problem, once the whole laser scan is divided into sets of consecutive range readings, a second segmentation criterion is applied into each set. This approach is focused on the correct selection of the set of dominant points present into a part of the scan bounded by two consecutive breakpoints. If the whole laser scan is divided into different sets of consecutive range readings by the breakpoint detector, this specific problem can be stated as the estimation of the curvature function associated to each set. Therefore, this one is based on the curvature associated to each range reading: consecutive range readings belong to the same segment while their curvature values are similar. To perform this segmentation task, the adaptive curvature function associated to each segment of the laser scan is computed [16]. Then, this information is employed to segment the laser scan into clusters of homogeneous curvature. The whole process to achieve this segmentation task is shown with details in [16]. Figure 5 (a) shows a real environment used to illustrate the CUBA framework. The scan data provided by the sensor and the curvature estimates by the segmentation stage are drawn in Figures 5 (b) and 5 (c), respectively.

Figure 4. (a)–(b) Laser scan and extracted breakpoints (squares). It must be noted that segments of the laser scan which present less than ten range readings are not taken into account (they are marked without boxes in the figures).



range readings to lines. This yields a nonlinear regression problem which can be solved for polar coordinates [33]. The line in the laser range finder's polar coordinate system is represented as

$$r = \frac{d}{\cos(\theta - \phi)} \quad (2)$$

where θ and d are the line parameters in the normal form representation:

$$x \cos \theta + y \sin \theta = d \quad (3)$$

being θ the angle between the X axis and the normal of the line and d the perpendicular distance of the line to the origin. Then, the orthogonal distance d_i of a range reading, $(r, \phi)_i$, to this line is

$$d_i = r_i \cos(\theta - \phi_i) - d \quad (4)$$

Under the assumption of known uncertainties, a weight for each measurement point can determine and fit the line in the generalized least squares sense, whose solution is (see [14] for further details)

$$\theta = \frac{1}{2} \arctan \left(\frac{\sum_i r_i^2 \sin 2\phi_i - \frac{2}{n} \sum_i \sum_j r_i r_j \cos \phi_i \sin \phi_j}{\sum_i r_i^2 \cos 2\phi_i - \frac{1}{n} \sum_i \sum_j r_i r_j \cos(\phi_i + \phi_j)} \right) \quad (5)$$

$$d = \frac{\sum_i r_i \cos(\phi_i - \theta)}{n}$$

Figure 5 (d) presents the detected landmarks corresponding to the scan data acquired by the sensor in 5 (b). In this case, Figure 5 (d) shows the line segments extracted using the described approach (end-points of the line segments are illustrated as squares). These end-points are determined by the intersection between this line and the two lines which are perpendicular to it and pass through the first and last range readings.

2. Curve segments

Although many circle fitting methods have been proposed, it is a common choice to achieve this by minimizing the mean square distance from the data points to the fitting circle. Basically, the Least Squares Fit (LSF) assumes that each data point is the noised version of the closest model point. This assumption is valid when data points are not contaminated with strong noise.

Let the data points be $\{x_i, y_i\}_{i=1 \dots m}$ ($m > 3$), with an uncertainty ellipse specified in terms of the standard deviations p_i and q_i , and the correlations r_i . The problem is to obtain the center (x_c, y_c) and the radius ρ of the circle C which yields the best fit to this data. It is also required to determine the variance matrix associated to the circle parameters.

This problem is stated as the minimization of the difference between the set of points $\{x_i, y_i\}$ and their corresponding points $\{x_c + \rho \cos \phi_i, y_c + \rho \sin \phi_i\}$ which lie on C . This difference is summarized by the $2m$ -element error vector ε :

$$\begin{aligned} \varepsilon &= (x_1 - (x_c + \rho \cos \phi_1), y_1 - (y_c + \rho \sin \phi_1), \dots, \\ & \quad x_m - (x_c + \rho \cos \phi_m), y_m - (y_c + \rho \sin \phi_m))^T \\ &= (\Delta x_1, \Delta y_1, \dots, \Delta x_m, \Delta y_m)^T \end{aligned} \quad (6)$$

This error vector has the known $2m \times 2m$ block diagonal variance matrix $V = \text{diag}(V_1 \dots V_m)$, where

$$V_i = \begin{bmatrix} p_i^2 & r_i \\ r_i & q_i^2 \end{bmatrix} \quad (7)$$

Then, assuming that the errors are normally distributed, the *maximum likelihood* (ML) problem consists of minimizing

$$\text{minimize } \varepsilon^T V^{-1} \varepsilon \quad (8)$$

with respect to the vector $b = (\phi_1, \dots, \phi_m, x_c, y_c, \rho)^T$.

In order to solve the minimization problem, the classical Gauss-Newton algorithm with the Levenberg-Marquardt correction [34, 35] is used. This algorithm finds the vector b which minimizes 8 in an iterative way. It approximates the objective function with the square of the norm of a linear function. Thus, at each iteration, the linear least-squares problem is solved

$$\text{min}_{\delta b} \|\varepsilon'^{(k)} - \nabla \varepsilon'^{(k)} \cdot \delta b\|_2 \quad (9)$$

where $\nabla \varepsilon'^{(k)}$ is the Jacobian matrix of first partial derivatives of ε' with respect to b and $\varepsilon'^{(k)}$ is ε' , both evaluated at $b^{(k)}$. A detailed description of the Levenberg-Marquardt algorithm can be found at [35]. In this case, the starting estimate for the centre coordinates and radius is obtained using the Taubin's approximation to the gradient-weighted algebraic circle fitting approach [34].

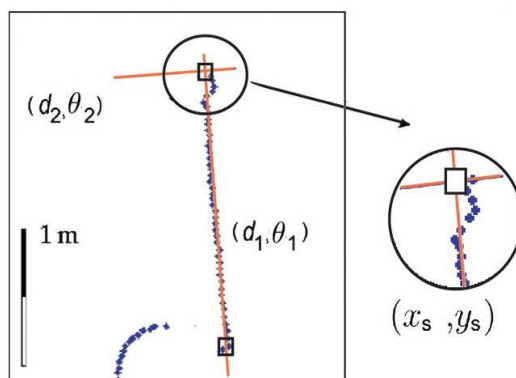
Finally, to obtain the variance matrix of the center coordinates and radius, an estimate of the variance matrix of the vector b must be obtained. Further details about the fitting problem are shown in [15]. Figure 5 (d) draws the circle segment extracted using the described approach for the scan data provided by the sensor in Figure 5 (b).

3. Real and virtual corners

As pointed out by Madhavan and Durrant-White [32], one of the main problems of a localization algorithm only based on corner detection is that the set of detected natural landmarks at each time step can be very limited, specially when it works on semi-structured environments. This generates a small observation vector that does not provide enough information to estimate the robot pose. To address this problem, the description algorithm described in this paper uses real and virtual corners as natural landmarks of the robot environment. Real corners are due to change of surface being scanned or change in the orientation of the scanned surface. Thus, they are not associated to laser scan discontinuities. On the other hand, virtual corners are defined as the intersection of extended line segments which are not previously defined as real corners. In order to obtain the corner location, it must be taken into account that failing to identify the correct corner point in the data can lead to large errors that increase with the distance to the detected corner (see Figure 6). Therefore, it is usually not a good option to locate the corner in one of the scan range readings. Another choice is to extract the corner location as the intersection of the two associated lines. Thus, the corner can be detected as the farthest point from a line defined by the two non-touching end-points of the lines or by finding that point in the neighborhood of the initial corner point, which gives the minimum sum of error variances of both lines [36]. The existence of a corner can be

determined from the curvature function [16], but its characterization (estimation of the mean pose and uncertainty measurement) is conducted using the two lines that generates the corner [14]. Figure 5 (d) illustrates the virtual corner detected by the algorithm (triangle) for the real scene described in Figure 5 (a). The associated covariance matrix has been also represented (ellipse).

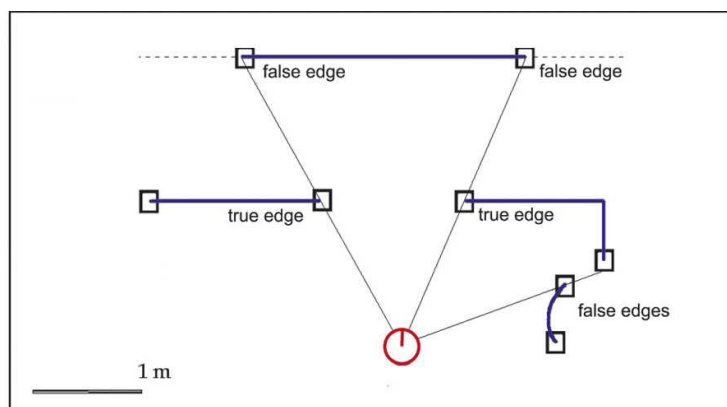
Figure 6. A real corner is not usually located at one of the laser range readings (they are marked as blue dots over the detected line segments).



4. Edges

The adaptive breakpoint detector searches for large discontinuity values in the laser scan data. Range readings that define this discontinuity are marked as breakpoints. Edges are defined as breakpoints associated to end-points of plane surfaces [37]. To satisfy this condition, the portion of the environment where the breakpoint is located must be a line segment and must not be occluded by any other obstacle. This last condition is true if the breakpoint is closer to the robot than the other breakpoint defined by the same large discontinuity (see Figure 7). It must be also noted that, when the laser range finder does not work with a scanning angle of 360° , the first and last breakpoints will not be considered as edges, because it is impossible to know if they define the end-point of a surface.

Figure 7. An edge is defined as a breakpoint associated to the end-point of a plane surface which is not occluded by any other obstacle.



Edges are characterized by the Cartesian position (x, y) of the breakpoint and by the orientation of the plane surface described by the line segment, θ ([14]).

4. Affine-invariant Laser Scan Segmentation

The CUBA algorithm is a curvature based approach to extract dominant points from the laser scan data. After the segmentation process a set of corners, line and curve segments are obtained and characterized to be used as natural landmarks. Although the adaptive curvature estimation provides a robust criterion to the laser scan segmentation, some aspects can be considered to improve this algorithm to be more robust against noise and affine transformations. This section describes the new technique proposed in this paper for segmenting the scan data acquired by a laser range sensor.

4.1. Adaptive Estimation of the Region-of-support

From the pioneering paper of Teh and Chin [38], many researchers have argued that the estimation of the curvature relies primarily on the precise calculation of the region-of-support associated to each curve point. In the described framework, in order to specify the region-of-support associated to the range reading i of the laser scan, the algorithm must determine the maximum length of scan that presents no significant discontinuities on the right and left sides of the range reading i , $t_f[i]$ and $t_b[i]$, respectively. To estimate the $t_f[i]$ value, the algorithm first computes two sets of triangles, $\{t_j^a\}_{j=i}^{i+t_f[i]-1}$ and $\{t_j^c\}_{j=i}^{i+t_f[i]-1}$. The area of the triangle t_j^a is defined as

$$|t_j^a| = \frac{1}{2} \begin{vmatrix} x_j & x_c & x_{j+1} \\ y_j & y_c & y_{j+1} \\ 1 & 1 & 1 \end{vmatrix} \quad (10)$$

where (x_j, y_j) and (x_{j+1}, y_{j+1}) are the Cartesian coordinates of the arc range readings j and $j + 1$ and (x_c, y_c) is the robot position, \mathbf{x}_c .

The area of the triangle t_j^c is defined as

$$|t_j^c| = \frac{1}{2} \begin{vmatrix} x_j^p & x_c & x_{j+1}^p \\ y_j^p & y_c & y_{j+1}^p \\ 1 & 1 & 1 \end{vmatrix} \quad (11)$$

where (x_j^p, y_j^p) is the projection of (x_j, y_j) on the chord that joins the range readings i and $i + t_f[i]$.

If $T_{i,t_f[i]}^a$ and $T_{i,t_f[i]}^c$ are equal to $\sum_{j=i}^{i+t_f[i]-1} |t_j^a|$ and $\sum_{j=i}^{i+t_f[i]-1} |t_j^c|$, respectively, then $t_f[i]$ will be defined by the largest value that satisfies

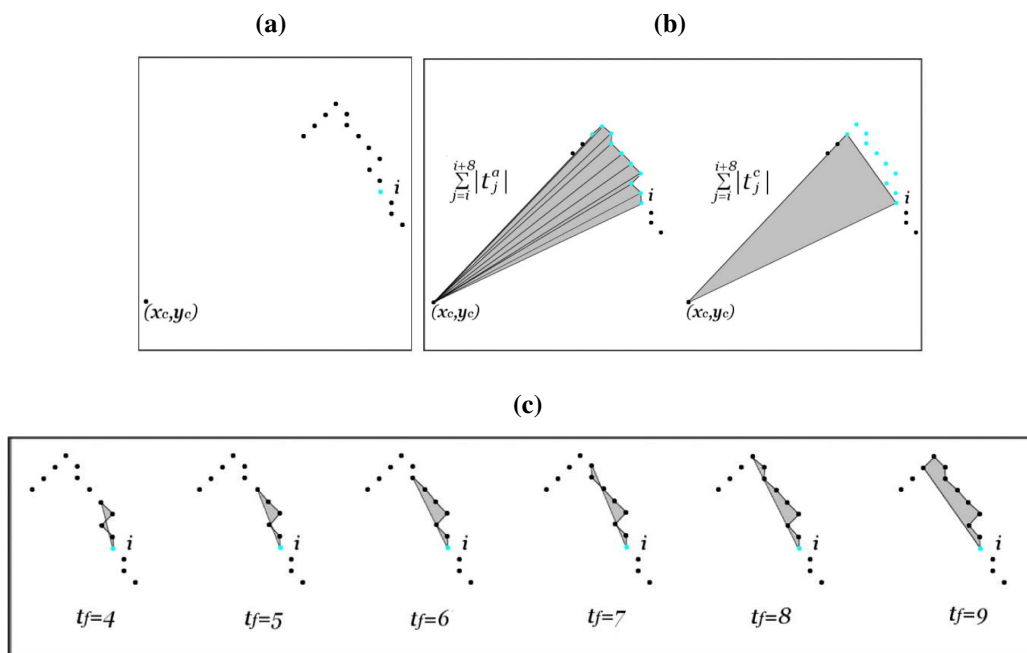
$$(T_{i,t_f[i]}^a - (T_{i,t_f[i]}^a \cap T_{i,t_f[i]}^c)) < U_t \quad (12)$$

Figure 8 shows the process to extract one $t_f[i]$ value. $t_b[i]$ is also set according to the described scheme, but using $i - t_b[i]$ instead of $i + t_f[i]$.

The correct selection of the U_t value is very important. Thus, if the value of U_t is large, $t_f[i]$ and $t_b[i]$ tend to be large and the contour details may be missed; if it is small, $t_f[i]$ and $t_b[i]$ are always very small and the resulting function is noisy. In order to set it correctly, a set of real plane surfaces have been

scanned at different distances from the sensor. In these surfaces, this value must be fixed to not to detect any local peak. This simple experiment has provided us an U_t value equal to 25.0 cm², which has been successfully employed in all experiments.

Figure 8. Calculation of the maximum length of contour presenting no significant discontinuity on the right side of range reading i ($t_f[i]$): (a) Part of the laser scan and point i ; (b) scan data acquired by the laser range sensor; and (c) evolution of the area delimited by the arc and the chord ($\sum_{j=i}^{i+t_f[i]-1} |t_j^a| - (\sum_{j=i}^{i+t_f[i]-1} |t_j^a| \cap \sum_{j=i}^{i+t_f[i]-1} |t_j^c|)$). It can be noted that this area suffers a sharp increasing when $t_f[i] \geq 8$. This change allows to estimate the correct $t_f[i]$ value and it will be detected in our approach using the Equation 12 (in this case, $t_f[i] = 8$).



4.2. Affine-invariant Laser Scan Segment Descriptor

Many researchers have used the area of the triangle, formed by the curve points, as the basis for shape representations [39]. The proposed laser scan segmentation algorithm employs a curvature estimator to characterize the shape contour, which is based on this triangle-area representation (TAR). Given a laser scan segment and, once this proposal has determined the local region-of-support associated to every range reading, the process to extract the associated TAR consists of the following steps:

1. Calculation of the local vectors \vec{f}_i and \vec{b}_i associated to each range reading i . These vectors present the variation in the X and Y axis between range readings i and $i + t_f[i]$, and between i and $i - t_b[i]$. If (x_i, y_i) are the Cartesian coordinates of the range reading i , the local vectors associated to i are defined as

$$\begin{aligned} \vec{f}_i &= (x_{i+t_f[i]} - x_i, y_{i+t_f[i]} - y_i) = (f_{x_i}, f_{y_i}) \\ \vec{b}_i &= (x_{i-t_b[i]} - x_i, y_{i-t_b[i]} - y_i) = (b_{x_i}, b_{y_i}) \end{aligned} \tag{13}$$

2. Calculation of the TAR associated to each range reading. The signed area of the triangle at contour point i is given by [39]:

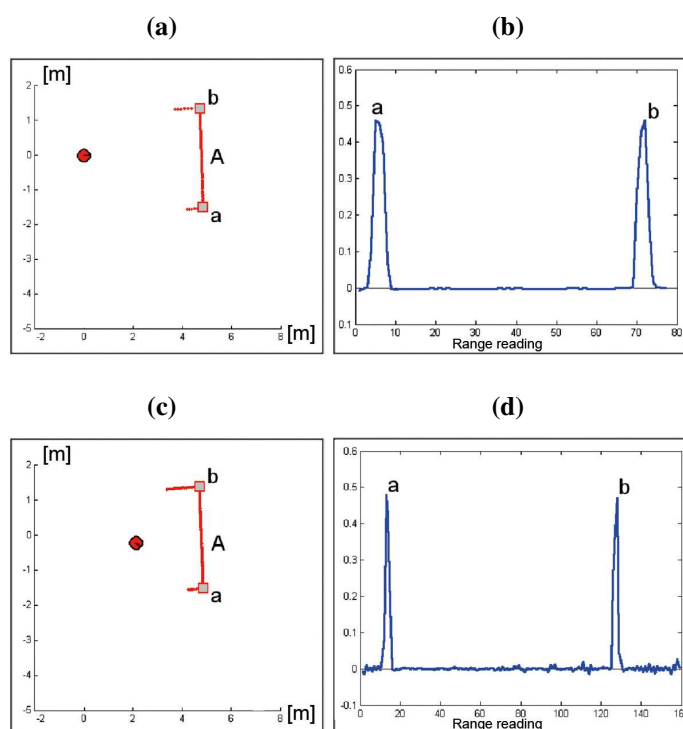
$$\kappa_i = \frac{1}{2} \begin{vmatrix} b_{x_i} & b_{y_i} & 1 \\ 0 & 0 & 1 \\ f_{x_i} & f_{y_i} & 1 \end{vmatrix} \quad (14)$$

3. TAR Normalization. The TAR of the whole laser scan segment, $\{\kappa_i\}_{i=1}^N$, is normalized by dividing it by its absolute maximum value.

When the contour is traversed counterclockwise, positive, negative and zero values of TAR mean convex, concave and straight-line points, respectively.

Figure 9 shows two laser scan segments taken from different points of view. Figures 9b and 9d present the two adaptive TAR associated to Figures 9a and 9c, respectively. Although the number of range readings in the acquired segments are significantly different, both representations detect the same sets of dominant points.

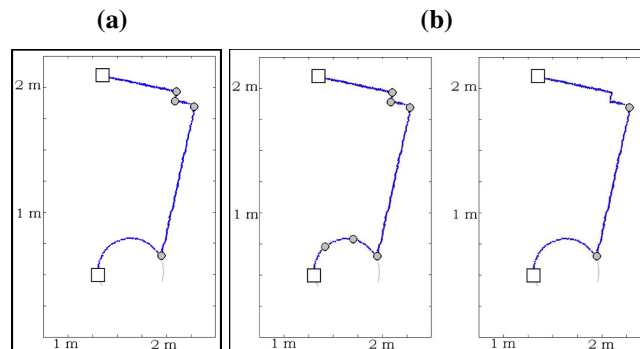
Figure 9. (a) Laser scan #1. (b) Adaptive TAR associated to scan segment A in (a). (c) Laser scan #2. (d) Adaptive TAR associated to scan segment A in (c).



The advantage of measuring the curvature in an adaptive way can be appreciated in Figure 10. Figure 10a shows the dominant points detected from the adaptive TAR associated to the laser scan segment (triangle side lengths ranging from 3 to 15). The scheme to detect line and curve segments described in [16] has been used. It can be noted that all dominant points are correctly detected. On the contrary, Figure 10a,b shows the dominant points obtained by the same process when two constant triangle side length values are used. It can be appreciated that when a low value is used ($t = 3$), the TAR is too

noisy and false dominant point detection occurs. On the contrary, if a high value is used ($t = 15$), the representation is excessively filtered, and some dominant points are lost.

Figure 10. (a) Dominant points detected from the TAR obtained using an adaptive triangle side length; and (b) dominant points detected from the TAR obtained using a t value equal to 3 (left image) or t value equal to 15 (right image).



4.3. Laser Scan Descriptor Under General Affine Transformations

Let $\{x_i, y_i\}_{i=1}^N$ be the Cartesian coordinates of the set of range readings associated to a laser scan segment. If this scan segment is subjected to an affine transformation, the relation between the original and the distorted representations is given by

$$\begin{bmatrix} \hat{x}_i \\ \hat{y}_i \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \quad (15)$$

where $\{\hat{x}_i, \hat{y}_i\}_{i=1}^N$ is the affine distorted representation of the scan segment, a , b , c and d represent scale, rotation and shear and t_1 and t_2 represent translation. By substituting the expressions for $\{\hat{x}_i, \hat{y}_i\}_{i=1}^N$ into Equation 13, we obtain

$$\begin{aligned} f_{\hat{x}_i} &= a(x_{i+t_f[i]} - x_i) + b(y_{i+t_f[i]} - y_i) = a \cdot f_{x_i} + b \cdot f_{y_i} \\ f_{\hat{y}_i} &= c(x_{i+t_f[i]} - x_i) + d(y_{i+t_f[i]} - y_i) = c \cdot f_{x_i} + d \cdot f_{y_i} \\ b_{\hat{x}_i} &= a(x_{i-t_b[i]} - x_i) + b(y_{i-t_b[i]} - y_i) = a \cdot b_{x_i} + b \cdot b_{y_i} \\ b_{\hat{y}_i} &= c(x_{i-t_b[i]} - x_i) + d(y_{i-t_b[i]} - y_i) = c \cdot b_{x_i} + d \cdot b_{y_i} \end{aligned} \quad (16)$$

Then, if we substitute 16 into 14, it is obtained that $\hat{\kappa}_i = (ad - bc)\kappa_i$, where $\hat{\kappa}_i$ is the affine transformed version of κ_i . As the TAR is normalized by its maximum value, this representation is invariant to the affine transformations.

4.4. Experimental Results

To evaluate the performance of the proposed algorithm, two laser scan datasets taken from two different environments have been selected: the fourth level of the University Institute of Research at the Technology Park of Andalusia in Málaga, Spain, and a part of the Intel Jones Farms Campus, Oregon. The first dataset has been collected using Rex, a Pioneer 2AT robot from ActivMedia equipped with a

SICK LMS200. The field of view is 180° in front of the robot and up to 8 m distance. The range samples are spaced every half a degree, all within the same plane. Figure 11a shows the first test area. This test area is an office-like environment which presents a higher density of detected landmarks. In the second test area, the scan data are taken in the Intel Jones Farms Campus in Oregon. The ground map is shown in Figure 11b. The environment is also an office-like structure. This second dataset has been obtained from the Radish repository [40] and the robot platform for this dataset was a Pioneer2DX (odometry) with a SICK LMS200. It must be noted that the set of threshold values used by the algorithm are the same for both scenarios.

Figure 11. (a) The first test area, an office-like environment sited at the Technology Park of Andalusia (Málaga); and (b) the map of a part of the Intel Jones Farms Campus in Hillsboro, Oregon (source: the Radish repository <http://radish.sourceforge.net/>).

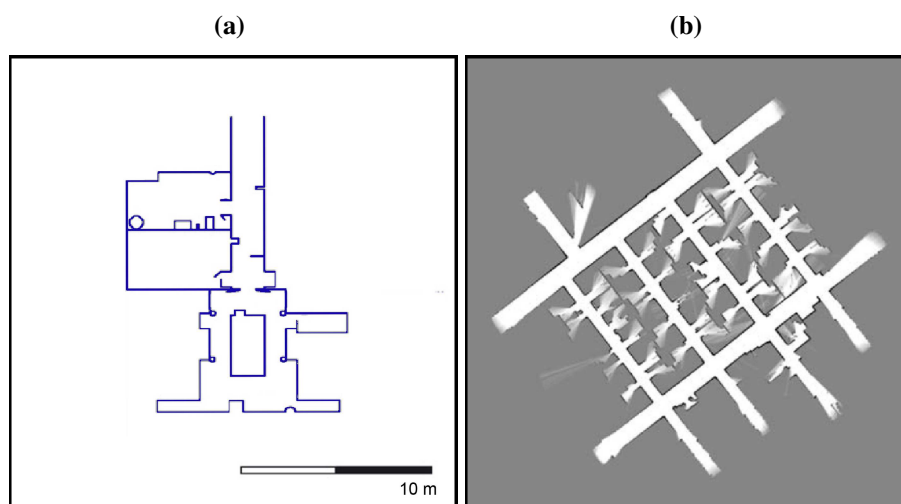


Figure 12 shows the detected segments at two different robot poses at the first scenario. Figures 12a and 12c present two scan data collected in these poses. The laser scan range readings has been marked as dots. The squares represent the start and end-points of each laser scan segment. Figures 12b and 12d show the curvature functions associated with the laser scans in Figures 12a and 12c, respectively. The segmented portions of the curvature functions are bounded by breakpoints or rupture points. Figure 13 shows the segmentation of several laser scans acquired by the robot in the second test area. It can be noted that all segments are correctly obtained.

Finally, in this work, due to the usual velocities in mobile robotics, the assumption of low speeds (i.e., few meters per second) has been taken. Therefore, the effect of the robot motion on individual range readings is negligible. Besides, the measured distance r_i is perturbed by a systematic error and a statistical error, usually assumed to follow a Gaussian distribution with zero mean. In order to compensate the systematic error, it has been approximated by a sixth-order polynomial which fits the differences between the measured distance and the true obstacle distance in the least-squares sense (see [14] for further details).

Figure 12. (a) Laser scan #3. (b) Curvature functions associated to (a). (c) Laser scan #4. (d) Curvature functions associated to (c).

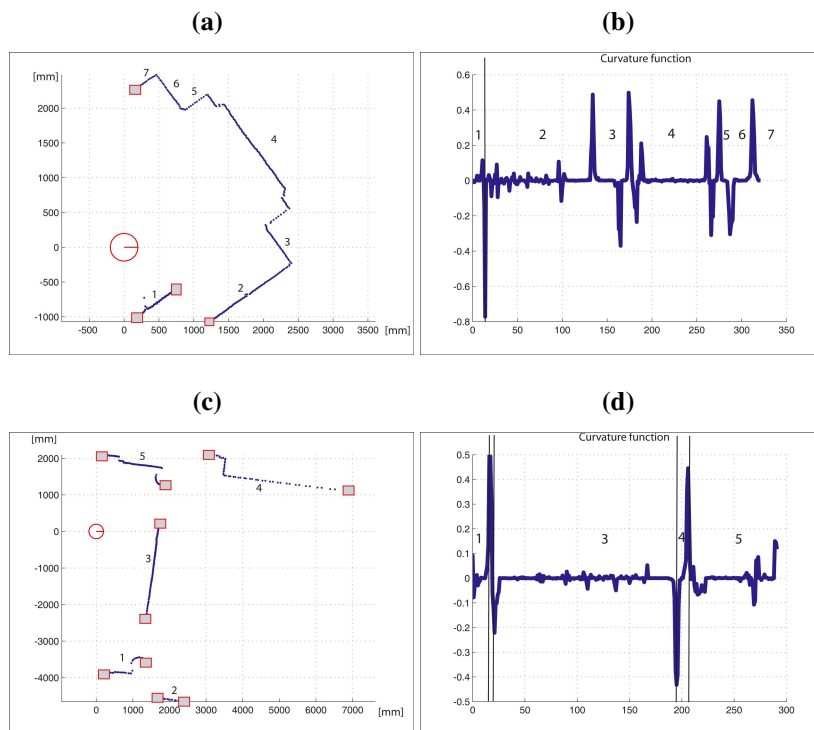
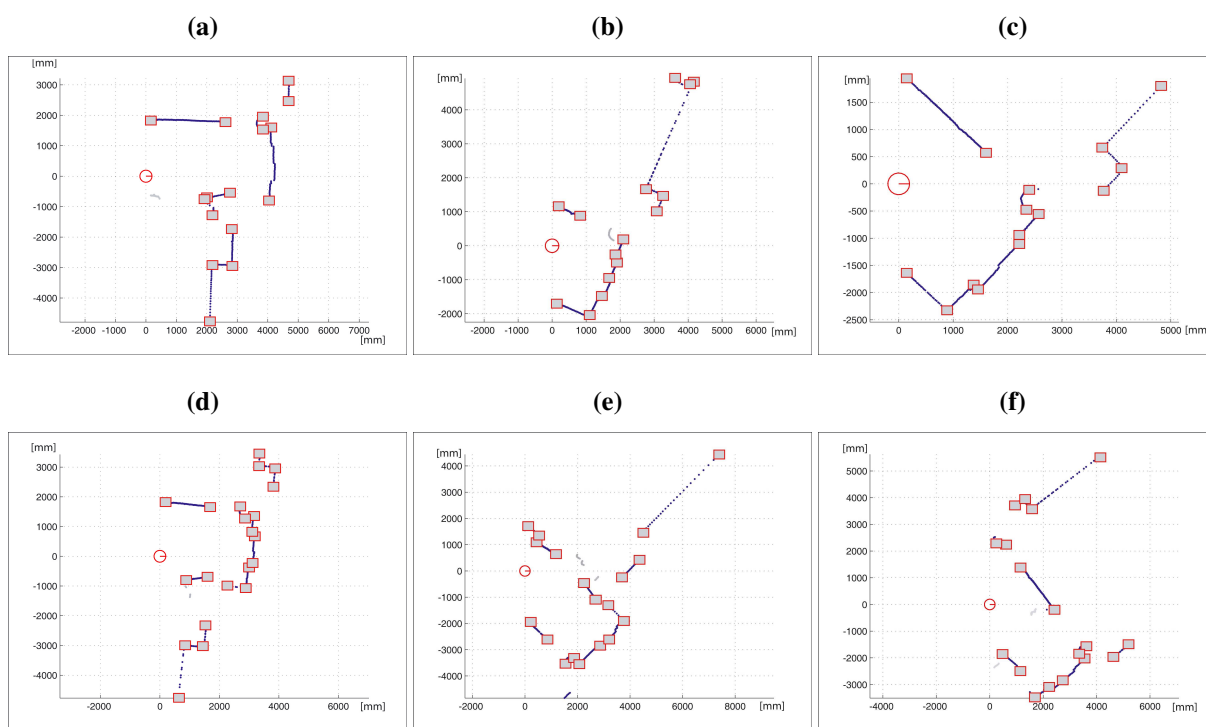


Figure 13. (a)–(f) Laser scan segmentations.



5. Comparative Study

In order to compare the proposed method to other approaches, we have implemented several laser scan data segmentation algorithms. Particularly, for the purpose of comparison, the split-and-merge (SM) algorithm [20], the iterative-end-point-fit (IEPF) method [18], the split-and-merge fuzzy (SMF) [19], the curvature scale space (CSS) [32], a Hough-based algorithm [23] and an adaptive curvature approach (CUBA) [16] has been selected. The test database consists of 50 laser scans obtained from a set of 10 artificial maps that have been created using the Mapper3 software from Activmedia Robotics. Laser scans have been obtained from these maps using MobileSim. The aim of using these artificial maps is to test each algorithm in a controlled and supervised environment, where the number and shape of segments are known (ground truth). Simulated laser sensor exhibits statistical errors of $\sigma_r = 5$ mm and $\sigma_\varphi = 0.1$ degrees. Each test scan consists of 360 range readings and it represents several line and curve segments.

Algorithms are programmed in C, and the benchmarks are performed on a PC with a Pentium II 450 MHz. The minimum number of points per line or curve segment have been fixed to 10 and the minimum physical length of a segment have been fixed to 50 cm. Both parameters have been chosen according to the simulated scans. Other parameters are algorithm-specific.

Segmentation experiments were repeated several times to choose good values for each approach. Segment pairs are initially matched using a χ^2 -test with a matching valid gate value of 2.77 (75% confidence interval). Then, extracted segments are matched to true segments using a nearest-neighbor algorithm. Experimental results are shown in Table 1. The correctness in these methods can be measured as [13]

$$TruePos = \frac{NumberMatches}{NumberTrueSeg} \quad (17)$$

$$FalsePos = \frac{NumberSegExAl - NumberMatches}{NumberSegExAl} \quad (18)$$

where *NumberSegExAl* is the number of segments extracted by an algorithm, *NumberMatches* is the number of matches to true segments and *NumberTrueSeg* is the number of true segments. To determine the precision, line and curve segments are taken into account. Line segments are characterized by α , the angle between the X axis and the normal of the line, and d , the perpendicular distance of the line to the origin. Then, the following two sets of errors on line parameters are defined:

$$\begin{aligned} \Delta d : \Delta d_i &= |d_i - d_i^t|, i = 1 \dots n \\ \Delta \alpha : \Delta \alpha_i &= |\alpha_i - \alpha_i^t|, i = 1 \dots n \end{aligned} \quad (19)$$

where n is the number of matched pairs, d_i^t and α_i^t are line parameters of a true line, and d_i and α_i are line parameters of the corresponding matched line. It is assumed that error distributions are Gaussian. Then, the variance of each distribution is computed as

$$\sigma_{\Delta d}^2 = \frac{1}{n-1} \sum (\Delta d_i - \frac{1}{n} \sum \Delta d_i)^2 \quad (20)$$

Similar sets of errors on curve parameters are defined (x_c and y_c define the center of the circle and ρ is the radius). From Table 1, it can be noted that the IEPF and SM algorithms perform faster than the others. The proposed method is faster than the CUBA, CSS and HT approaches. Besides, the CSS, CUBA and

the proposed algorithm are the only methods that do not split curve segments into short straight-line segments. Therefore, they have the best scores in term of correctness and precision with respect to curve segments.

Finally, a typical situation to illustrate the improvement in the proposed algorithm is shown in Figure 14. In this case it is compared to the CUBA algorithm (Figures 14c and 14d) in the same indoor environment but from different robot poses. The inset of Figure 14d shows a part of the scan in detail, where the algorithm split the curve segment in two parts. This is not the result obtained from the previous pose (Figure 14c), in this case the same part of the scan has been considered as only one curve segment. It can be noted how the proposed algorithm provides the same results from different poses (Figures 14a and 14b) due to its invariant properties. These situations arise when some parts of the environment are occluded or are out of the vision field of the laser scanner during some time, and are observed again from a different robot pose (translation and/or rotation).

Figure 14. (a)–(b) Results of the proposed algorithm from different poses in the same indoor environment. The same results are provided by the algorithm from these robot poses (see details in the figure (b)); (c)–(d) CUBA algorithm results from the same tests. The algorithm split the curve segment in different parts due to this algorithm is not invariant to robot pose.

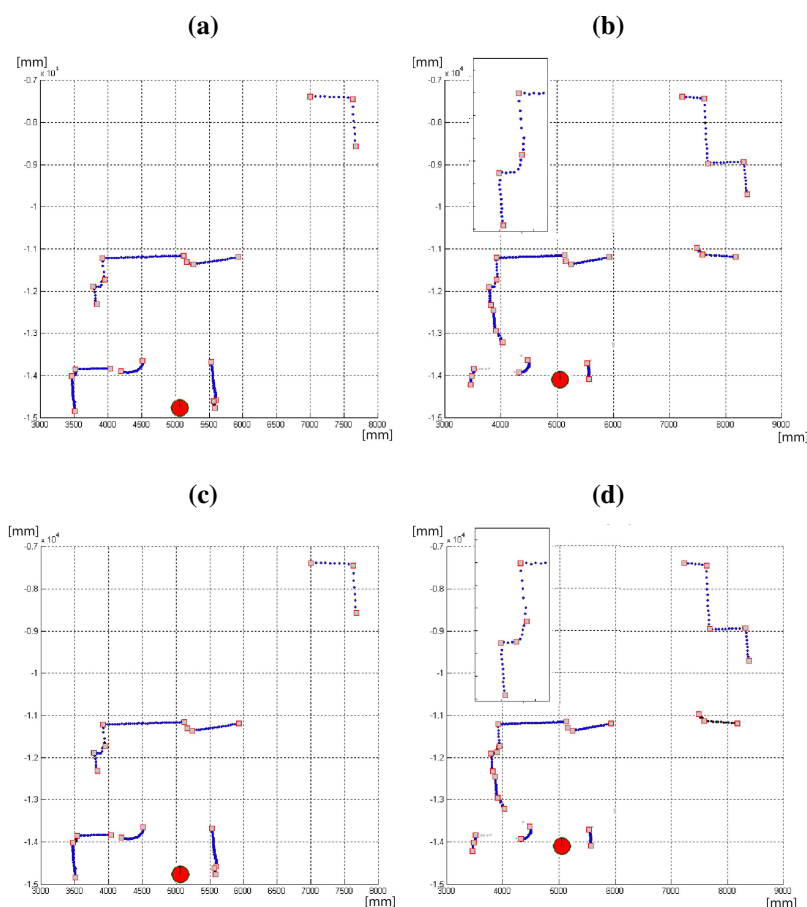


Table 1. Experimental results of several segmentation algorithms (see text).

Algorithm	SM	SMF	IEPF	CSS	HT	CUBA	Proposed
Execution time (ms)	7.1	14.6	4.2	39.1	33.4	13.6	10.1
TruePos	0.77	0.79	0.78	0.93	0.73	0.92	0.93
FalsePos	0.23	0.23	0.26	0.03	0.28	0.02	0.02
$\sigma_{\Delta d}$ [mm]	15.2	12.3	17.2	10.4	9.8	10.2	10.2
$\sigma_{\Delta \alpha}$ [deg]	0.82	0.63	0.79	0.58	0.58	0.60	0.59
$\sigma_{\Delta x_c}$ [mm]	14.1	13.1	13.9	10.1	13.0	9.8	9.7
$\sigma_{\Delta y_c}$ [mm]	12.6	12.9	13.1	9.8	12.7	9.9	9.6
$\sigma_{\Delta \rho}$ [mm]	8.3	7.9	8.5	7.9	8.5	6.5	6.1

6. Conclusions and Future Works

This paper presents a curvature-based environment description for robot navigation using laser range sensors. The main advantage of using curvature information is that the algorithm can directly provide line and curve segments, and improve the robustness against noise. Besides, this approach exhibits superior performance over traditional segmentation algorithms based on polygonal approximations which assume that the laser scan is only composed of line segments. The proposed segmentation module uses an adaptive estimate of the curvature according to a triangle-area representation, where the triangle side length at each range reading are adapted to the local changes of the scan data. The segmentation results are used to provide natural landmarks of the robot environment: line and curve segments, corners and edges. Finally, the segmentation algorithm has been compared with the state-of-the-art algorithms in terms of performance, robustness and speed. Although the proposed algorithm provides slightly better results than previously proposed curvature-based approaches (the best results has been obtained for curve segments, see Table 1), it is faster, invariant to translation and rotation, and robust against noise. Future work will focus on the development of an algorithm for scan matching of two consecutive scan data acquired by the sensor based on the extracted features and application in dynamic environments (e.g., people or object moving around the robot). This algorithm must be capable to estimate the robot pose by trying to maximize the overlapping between two sets of features extracted by the environment description proposed in this work.

7. Glossary

1. *Odometry*: is the use of data from the movement of proprioceptive sensor (actuators) to estimate change in the robot pose over time. Odometry is used by the current robots to estimate (not determine) their pose relative to an initial location.
2. *Localization*: is defined as the knowledge of the position and orientation of the robot in the working environment at every instant of time. In a local point of view, given a map of the environment and an initial pose (x - y position, and θ orientation), the localization task consists of tracking the

mobile agent around the environment.

3. *Mapping*: is defined as the problem of acquiring a spatial model of a robot environment. Usually, mapping algorithms obtain an instantaneous local representation of the scene according to the current sensor reading, including static and dynamic objects. Next, a global map is built only with static objects.
4. *SLAM*: is a technique used by autonomous mobile robots to build up a map within an unknown environment while keeping track of their current position at the same time.
5. *Natural landmarks*: Landmarks are defined as features which are determined by the system and detected according to some criteria. In this situation, natural landmarks are directly selected in the natural scene considering their geometrical or photo-metrical features.
6. *Segmentation*: is a process of aiming to classify each scan data into several groups, each of which possibly associates with different structures of the environment.
7. *Breakpoints*: are scan discontinuities due to a change of the surface being scanned by the laser sensor.

Acknowledgements

This work has been partially granted by the Spanish Ministerio de Ciencia e innovación (MCINN) and FEDER funds, and by Junta de Andalucía, under projects no. TIN2008-06196 and P07-TIC-03106, respectively. The Intel Oregon dataset was obtained from the Robotics Data Set Repository (Radish). Thanks go to Maxim Batalin for providing the data.

References and Notes

1. Roumeliotis, S.; Bekey, G.B. Segments: A Layered, Dual-Kalman Filter Algorithm for Indoor Feature Extraction. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kagawa University, Takamatsu, Japan, November 5 - October 30, 2000; pp. 454–461.
2. Tardós, J.; Neira, J.; Newman, P.; Leonard, J. Robust mapping and localization in indoor environments using sonar data. *Int. J. Robot. Res.* **2002**, *40*, 311–330.
3. Crowley, J. World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, USA, May 14-19, 1989; pp. 674–680.
4. Gutmann, J.; Schlegel, C. AMOS: Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*, Kaiserslautern, Germany, October 9-11, 1996; pp. 61–67.
5. Yagub, M.T.; Katupitaya, J. Line Segment Based Scan Matching for Concurrent Mapping and Localization of a Mobile Robot. In *Proceedings of the 2006 IEEE International Conference on Control, Automation, Robotics and Vision*, Grand Hyatt, Singapore, December 5-8, 2006; pp. 1–6.

6. Lingemann, K.; Nuchter, J.H.; Surmann, H. High-speed laser localization for mobile robots. *Int. J. Robot. Auton. Syst.* **2005**, *51*, 275–296.
7. Kosaka, A.K. Fast Vision-guided Mobile Robot Navigation Using Model-based Reasoning And Prediction Of Uncertainties. In *Proceedings of the 1992 IEEE International Conference on Intelligent Robots and Systems*, Raleigh, USA, July 7-10, 1992; pp. 2177–2186.
8. Ayache, N.; Faugeras, O. Maintaining representations of the environment of a mobile robot. *IEEE Trans. Robot. Autom.* **1989**, *5*, 804–819.
9. Atiya, S.; Hager, G. Real-time vision-based robot localization. *IEEE Trans. Robot. Autom.* **1993**, *9*, 785–800.
10. Se, S.; Lowe, D.; Little, J. Vision-based Mobile Robot Localization and Mapping Using Scale-Invariant Features. In *IEEE Proceedings of the 2001 International Conference on Robotics and Automation*, Seoul, Korea, May 21-26, 2001; pp. 2051–2058.
11. Martin, M. Evolving visual sonar: Depth from monocular images. *Patt. Rec. Lett.* **2006**, *27*, 1174–1180.
12. Xu, K.; Luger, G. The model for optimal design of robot vision systems based on kinematic error correction. *Image Vis. Comp.* **2007**, *25*, 1185–1193.
13. Nguyen, V.; Gächter, S.; Martinelli, A.; Tomatis, N.; Siegwart, R. A comparison of line extraction algorithms using 2D range data for indoor mobile robotics. *Auton. Rob.* **2007**, *23*, 97–111.
14. Núñez, P.; Vázquez-Martín, R.; del Toro, J.; Bandera, A.; Sandoval, F. Natural landmark extraction for mobile robot navigation based on an adaptive curvature estimation. *Robot. Auton. Syst.* **2008**, *56*, 247–264.
15. Núñez, P.; Vázquez-Martín, R.; Bandera, A.; Sandoval, F. An algorithm for fitting 2-D data on the circle: applications to mobile robotics. *IEEE Sig. Proc. Lett.* **2008**, *15*, 127–130.
16. Núñez, P.; Vázquez-Martín, R.; del Toro, J.; Bandera, A.; Sandoval, F. Feature Extraction from Laser Scan Data Based on Curvature Estimation for Mobile Robotics. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, USA, May 15-19, 2006; pp. 1167–1172.
17. Castellanos, J.; Tardós, J. Laser-based segmentation and localization for a mobile robot. In *Robotics and Manufacturing: Recent Trends in Research and Applications 6*; Jamshidi, M., Pin, F., Dauchez, P., Eds.; ASME Press: New York, NY, USA, 1996.
18. Zhang, L.; Ghosh, B.K. Line Segment Based Map Building and Localization Using 2D Laser Rangefinder. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, USA, April 24-28, 2000; pp. 2538–2543.
19. Borges, G.; Aldon, M. Line extraction in 2D range images for mobile robotics. *J. Int. Robot. Syst.* **2004**, *40*, 267–297.
20. Nguyen, V.; Martinelli, A.; Tomatis, N.; Siegwart, R. A Comparison of Line Extraction Algorithms Using 2D Laser Rangefinder for Indoor Mobile Robotics. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Alberta, Canada, August 2-5, 2005; pp. 1929–1934.
21. Iocchi, L.; Nardi, D. Hough localization for mobile robots in polygonal environments. *Rob. Auton. Syst.* **2002**, *40*, 43–58.

22. Pfister, S.; Roulletiotis, S.; Burdick, J. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 14-19, 2003; pp. 1304–1311.
23. Bandera, A.; Pérez-Lorenzo, J.; Bandera, J.; Sandoval, F. Mean shift based clustering of Hough domain for fast line segment detection. *Patt. Recog. Lett.* **2006**, *27*, 578–586.
24. Martínez-Cantin, R.; Castellanos, J.; Tardós, J.; Montiel, J. Adaptive Scale Robust Segmentation for 2D Laser Scanner. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 9-15, 2006; pp. 796–801.
25. Mokhtarian, F.; Mackworth, A. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. Patt. Anal. Machine Intell.* **1986**, *8*, 34–43.
26. Fontoura, L.; Marcondes, R. Shape analysis and classification. In *Shape Analysis and Classification*. CRC Press: Boca Raton, FL, USA, 2001.
27. Agam, G.; Dinstein, I. Geometric separation of partially overlapping nonrigid objects applied to automatic chromosome classification. *IEEE Trans. Patt. Anal. Machine Intell.* **1997**, *11*, 1211–1222.
28. Liu, H.; Srinath, D. Partial shape classification using contour matching in distance transformation. *IEEE Trans. Patt. Anal. Machine Intell.* **1990**, *11*, 1072–1079.
29. Arrebola, F.; Bandera, A.; Camacho, P.; Sandoval, F. Corner detection by local histograms of the contour chain code. *Elect. Lett.* **1997**, *33*, 1769–1771.
30. Bandera, A.; Urdiales, C.; Arrebola, F.; Sandoval, F. Corner detection by means of adaptively estimated curvature function. *Elect. Lett.* **2000**, *36*, 124–126.
31. Reche, P.; Urdiales, C.; Bandera, A.; Trazegnies, C.; Sandoval, F. Corner detection by means of contour local vectors. *Elect. Lett.* **2002**, *38*, 699–701.
32. Madhavan, R.; Durrant-Whyte, H. Natural landmark-based autonomous vehicle navigation. *Robot. Auton. Syst.* **2004**, *46*, 79–95.
33. Arras, K.; Siegwart, R. Feature Extraction and Scene Interpretation for Map Based Navigation and Map Building. In *Proceedings of SPIE, Mobile Robotics XII, vol 3210*, Pittsburgh, USA, October 14-17, 1997; pp. 42–53.
34. Chernov, N.; Lesort, C. Least squares fitting of circles. *J. Math. Imag. Vision* **2005**, *23*, 239–251.
35. Shakarji, C. Least-squares fitting algorithms of the NIST algorithm testing system. *J. Res. Natl. Inst. Stand. Tech.* **1998**, *103*, 633–641.
36. Diosi, A.; Kleeman, L. Uncertainty of line segments extracted from static sick pls laser scans. Technical Report MECSE-26-2003, Department of Electrical and Computer Systems Engineering, Monash University, 2003.
37. Zhang, S.; Xie, L.; Adams, M. Feature extraction for outdoor mobile robot navigation based on a modified Gauss-Newton optimization approach. *Robot. Auton. Syst.* **2006**, *54*, 277–287.
38. Teh, C.; Chin, R. On the detection of dominant points on digital curves. *IEEE Trans. Patt. Anal. Machine Intell.* **1989**, *11*, 859–872.
39. Alajlan, N.; Rube, I.E.; Kamel, M.; Freeman, G. Shape retrieval using triangle-area representation and dynamic space warping. *Patt. Recog.* **2007**, *40*, 1911–1920.

40. Howard, A.; Roy, N. The robotics data set repository (radish). Available online: <http://radish.sourceforge.net/>.

© 2009 by the authors; licensee Molecular Diversity Preservation International, Basel, Switzerland. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).