Available online at www.sciencedirect.com
ScienceDirect
Robotics and Autonomous Systems 56 (2008) 247–264
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Natural landmark extraction for mobile robot navigation based on an adaptive curvature estimation☆

P. Núñez*, R. Vázquez-Martín, J.C. del Toro, A. Bandera, F. Sandoval

Grupo de Ingeniería de Sistemas Integrados, Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Universidad de Málaga, Campus de Teatinos s/n 29071-Málaga, Spain

Received 28 November 2006; received in revised form 16 July 2007; accepted 25 July 2007
Available online 31 July 2007

Abstract

This paper proposes a geometrical feature detection system which is to be used with conventional 2D laser range finders. It consists of three main modules: data acquisition and pre-processing, segmentation and landmark extraction and characterisation. The novelty of this system is a new approach for laser data segmentation based on an adaptive curvature estimation. Contrary to other works, this approach divides the laser scan into line and curve segments. Then, these items are used to directly extract several types of landmarks associated with real and virtual features of the environment (corners, center of tree-like objects, line segments and edges). For each landmark, characterisation provides not only the parameter vector, but also complete statistical information, suitable to be used in a localization and mapping algorithm. Experimental results show that the proposed approach is efficient to detect landmarks for structured and semi-structured environments.

Keywords: Natural landmark extraction; Mobile robot navigation; Adaptive curvature estimation; Covariance matrix

1. Introduction

Reliable navigation is a fundamental competence for autonomous mobile robotics. The basic idea behind most of the current navigation systems operating in a known environment is that the robot carries sensors to perceive the environment and match the obtained data with the expected data available in a previously generated map. The robot uses this operation to update its pose—position and orientation—and correct the localization error due to odometry slippage. In addition, sensor information can be used to simultaneously localize the robot and build the map of the environment along the robot's trajectory. The difficulty of simultaneous localization and map building (SLAM) problem lies in the fact that an accurate estimation of the robot trajectory is required to obtain a good map, and for reducing the unbounded growing odometry errors requires to associate sensor measurements with a precise map [30]. The SLAM problem has received considerable attention over the last decade and different solutions have been proposed. In order to increase the efficiency and robustness of the process, sensor data have to be transformed in a more compact form before attempting to compare them to the ones presented on a map or store them in a simultaneously built map. In either case, the chosen map representation heavily determines the precision and reliability of the whole task [28]. Typical choices for the map representation include topological [18], cell-based [14], feature- or landmark-based models [30] and sequential Monte Carlo methods [31]. In this paper, we adopt a feature-based approach for the map representation, where landmarks can be defined as "*distinct features that a vehicle can recognize reliably from its sensor observations*" [20]. These approaches allow the use of multiple models to describe the measurement process for different parts of the environment and avoid the data smearing effect [30]. However, the success of this representation is conditioned on

- the chosen type of landmark and the existence of accurate sensor capable of discriminating between similar landmarks; and
- the availability of fast and reliable algorithms capable of extracting landmarks from a large set of noisy and uncertain data.

☆ This work has been partially funded by Spanish Ministerio de Educación y Ciencia, MEC and FEDER Fund, project nº. TIN2005-01359.
* Corresponding author. Tel.: +34 952137222; fax: +34 952131447.
E-mail address: pmnt@uma.es (P. Núñez).

0921-8890/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2007.07.005

Thus, in most of these algorithms, it is decisive to choose a correct type of landmark. This decision can be made according to the environment where the robot is moving and the external sensor used to obtain the data. Regarding this question, visual landmarks can be used in indoor environments and outdoors, e. g. Harris corners [12] or SIFT (Scale Invariant Feature Transform) features [17]. However, if a range sensor is employed, features such as walls or corners are used as landmarks in structured environments [1,28]. In unstructured or natural outdoor environments, similar simple landmarks can be very infrequently detected. Therefore, some authors have proposed to use tree trunks or tree-like objects as naturally occurring landmarks [16,36]. In our case, as several authors have pointed out [36,20], it is assumed that structured and semi-structured environments have common features (tree trunks, columns, corners or walls) which can be described by items as line segments, corners or curve segments in a planar representation. With respect to the selected sensor to perceive these landmarks, sonar, laser or vision-based systems sensors are commonly used. Applying vision to feature extraction leads to increase CPU usage due to the complexity of the algorithms required. On the contrary, the complexity of feature extraction algorithms that work with sonar or laser sensors is usually very reduced. If we assume that the structural features commonly found in the environment are invariant to height (e.g. walls, corners, columns), a planar representation would be adequate for feature extraction. Sonar sensors suffer from frequent specular reflections and a significant spread of energy (beamwidth). Instead, a laser range scanner is capable of collecting such high quality range data and it suffers from very small number of specular reflections. The angular uncertainty of the laser sensor is very small and, therefore, it can provide a very fine description of the robot's surroundings. Finally, although from the perspective of cost, laser scanners are more expensive than sonar sensors, it can be appreciated that it is an affordable device for most mobile robotics systems.

On the other hand, pattern recognition concepts and algorithms can be applied to extract features from sensor data. Thus, simple methods have been broadly used to support mobile robot operation using line or point features extracted from range images [35,10]. Although these methods are very fast, they have problems in dealing with adverse phenomena such as false measurements on surface limits [9]. Besides, they do not consider sensor motion. More robust methods that take into account sensor motion have been also proposed [4,25]. These methods are based on more elaborate concepts, like the Hough transform [4], the fuzzy clustering [9] or the Kalman filter [28]. The main drawbacks of the majority of these methods are they only look for one type of feature (e.g., line segment). On the contrary, a concept that can be used to obtain several types of features from the laser scan is the local curvature value [20,24]. Besides, features extracted from local curvature are view-point invariant measures and, this means that they can be used as robust landmarks in localization. In [20], an iterative curvature scale space (CSS) approach is used to detect corners from the laser scan data. In [24], corners, line segments and curve segments are simultaneously extracted from the laser scan using an adaptively estimated curvature function.

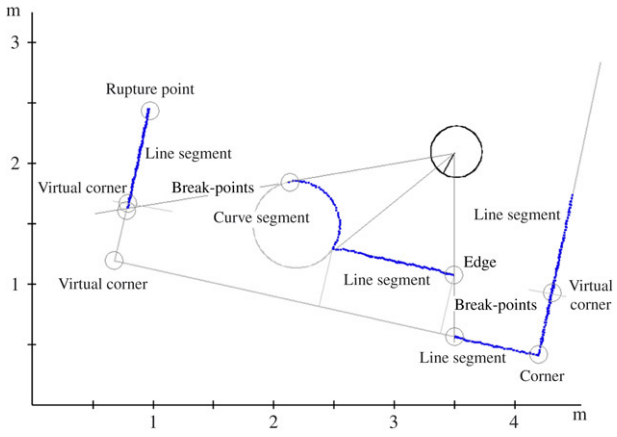

Fig. 1. Sensor information obtained from a single laser scan using a SICK laser scanner.

The aim of this work is to extract and characterise several types of landmarks that are present in structured and semi-structured environments using a laser range finder (see Fig. 1). The approach must be fast and capable of extracting landmarks from noisy and uncertain data. Moreover, characterisation must provide not only geometrical information, but also complete statistical information to be used in later navigation applications as SLAM. Particularly, the laser scan is analyzed to detect rupture points and breakpoints [9] and four types of landmarks: line segments, corners, center of curve segments and edges. Such items are collected from the environment as follows:

- Rupture points are scan measurements associated with discontinuities due to the absence of obstacles in the scanning direction.
- Breakpoints are scan discontinuities due to the change of surface being scanned by the laser sensor.
- Line segments result from the scan of planar surfaces (e.g. walls).
- Real corners are due to the change of surface being scanned or due to the change in the orientation of the scanned surface. Corners are not associated with laser scan discontinuities.
- Virtual corners are defined by the intersection of two lines corresponding to previously detected line segments. Thus, they do not correspond to any real corner of the environment.
- Center of curve segments result from the scan of curve surfaces (e.g. trees or cylindrical columns).
- Edges are defined as breakpoints associated with free end-points of plane surfaces [36]. They are also called *semiplanes* [11]. Edges might correspond to frames of open doors or convex corners and their detection must be carefully achieved to avoid false edges arising from occlusions [11].

In this paper, we present a geometrical feature detection framework for using with conventional 2D laser sensors. This framework is composed of three procedures [10,9]: data acquisition and pre-processing, laser scan segmentation and landmark extraction and characterisation, which are described in the rest of the paper. Thus, Section 2 shows the characteristics

of the laser sensor and the data pre-processing. Section 3 presents the laser scan data segmentation which consists of two stages. Firstly, the adaptive breakpoint detection method [9] is used to look for large discontinuities. Then, the adaptive curvature function is used to segment the whole scan data into clusters of range readings which present a homogeneous curvature value. Section 4 describes the detection and characterisation of the different types of landmarks. Section 5 presents the experimental results and Section 6 compares the proposed method with other similar approaches. Finally, Section 7 summarizes conclusions and future work.

2. Laser scan data acquisition and pre-processing

The information provided by laser sensors in a single scan is usually quite dense and has good angular precision. Range images provided by laser range finders are typically in the form $\{(r, \phi)_{l|l=1\ldots N_R}\}$, on which $(r, \phi)_l$ are the polar coordinates of the lth range reading (r_l is the measured distance of an obstacle to the sensor rotating axis at direction ϕ_l) and N_R is the number of range readings related to the angular range of the measurement R and the laser angular resolution $\Delta\phi$, through $N_R = \frac{R}{\Delta\phi}$. The scan measurements are acquired by the laser range finder with a given angular resolution $\Delta\phi = \phi_l - \phi_{l-1}$. The distance r_l is perturbed by a systematic error, ϵ_s, and a statistical error, ϵ_r, usually assumed to follow a Gaussian distribution with zero mean and variance σ_r^2. Then, if r_m is the measured distance and r_t the true obstacle distance, it can be considered that they are related by

$$r_m - r_t = \epsilon_s(r_m) + \epsilon_r. \tag{1}$$

Our laser range finder is a SICK Laser Measurement System (LMS) 200, and the experiments have been performed with the LMS doing planar range scans with angular range of 180° operating at frequencies of about 60 Hz, and maximum measurement range of 8 m. In these conditions, the SICK LMS200 laser sensor exhibits a systematic error of ±15 mm and a statistical error (σ_r) of 5 mm. Taken several values of r_m for $r_t \in [0.1, 8]m$, the systematic error $\epsilon_s(r_m)$ can be easily approximated by a sixth-order polynomial which fits the differences $r_m - r_t$ in the least-squares sense [9]. This polynomial is used for compensating the systematic error according to the model (1). The residual noise after systematic error correction is compatible with the value $\sigma_r = 0.005$ m provided by the laser range finder manufacturer. Besides, although several authors omit the term σ_ϕ which keeps track of angular uncertainties [3,13], we have assumed that this term can reach its maximum value, i.e. $\sigma_\phi = \Delta\phi/4$, where $\Delta\phi$ is the laser angular resolution. When range images are taken with the robot in motion, they may be deformed during the scanning time. In such cases, a compensation algorithm based on estimates of the robot motion should be applied. In our system, the motion correction algorithm described in [4] is used. Basically, the vehicle displacement during a scan is compensated by transforming each range reading acquired at instant time t_l to the desired reference time t_1. Let $\{(x, y)_{l|l=1\ldots N_R}\}$ be the Cartesian representation of the range images, where $x_l = r_l \cos\phi_l$ and $y_l = r_l \sin\phi_l$, and $p_l = (x_s, y_s, \theta_s)_l$ the sensor absolute position when the lth range reading is acquired. At the lth range reading acquisition, the local coordinate frame has been displaced $p_d^l = p_l - p_1$ from the start of range reading acquisition. In order to recover the coordinates of the lth range reading when the sensor is on p_1, $(x_l^1, y_l^1, \theta_l^1)$, the sensor displacement is taken into account as

$$\begin{pmatrix} x_l^1 \\ y_l^1 \end{pmatrix} = \begin{pmatrix} \cos\theta_d^l & \sin\theta_d^l \\ -\sin\theta_d^l & \cos\theta_d^l \end{pmatrix} \cdot \begin{pmatrix} x_l + x_d^l \\ y_l + y_d^l \end{pmatrix}. \tag{2}$$

Thus, it is not necessary to know the sensor absolute pose at each lth point, only its relative displacement. In our experiments, it is assumed that odometry can provide a good estimation of this movement. In fact, the operating frequency of the laser range finder is very high and the sensor displacement p_d^l is interpolated by a linear relation between p_d^1 and $p_d^{N_R}$. These values can be derived from odometry in t_1 and t_{N_R}.

Finally, at the same time that the systematic error and the motion are corrected, rupture points can be detected. A rupture point is defined as a discontinuity during the laser measurement and can be due to absence the of obstacles in that direction. SICK LMS200 returns a predefined binary data to indicate this occurrence.

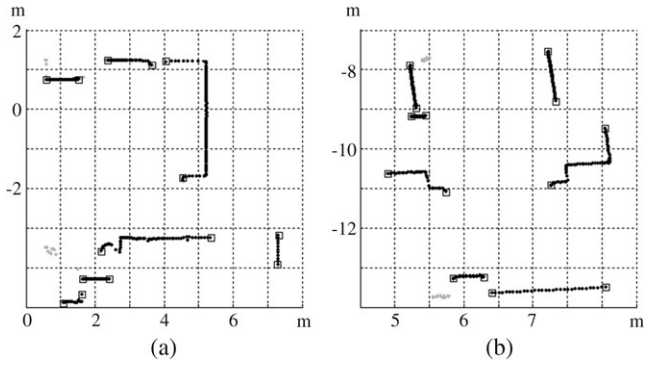

Fig. 2. (a)–(b) Laser scan and extracted breakpoints (squares). It must be noted that segments of the laser scan which present less than ten range readings are not taken into account (they are marked as grey colored range readings).

3. Laser scan data segmentation

Segmentation is a process whose aim is to classify each scan data into several groups, each one of them is associated with different surfaces of the environment. In our approach, the segmentation is achieved in two consecutive steps. Firstly, scan data is segmented using the adaptive breakpoint detector [9]. This algorithm permits to reject isolated range readings, but it provides an undersegmentation of the laser scan, i.e. extracted segments between breakpoints typically group two or more different structures (see Fig. 2). In order to avoid this problem, a second segmentation criterion is applied to each segment. This one is based on the curvature associated with 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 with each segment of the laser scan is obtained [24].

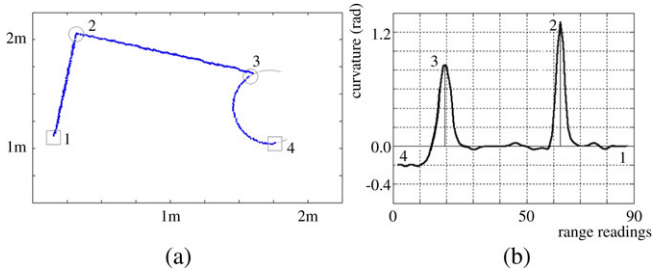

Fig. 3. (a) Segment of a single laser scan (□—breakpoints, o—corners); and (b) curvature function associated with (a).

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. Fig. 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 (Fig. 3(b)). It also shows that segment 3–4 has an average value larger than zero, but it is not flat due to noise. Nevertheless, peaks in that segment are too low to be considered corners of the curve. Finally, segments 1–2 and 2–3 present a curvature average value near to zero, as is expected in line segments.

In a general case, the curvature $\kappa(t)$ of a parametric plane curve, $c(t) = (x(t), y(t))$, can be calculated as [21,15]

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

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 [15]. 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, they differentiate the interpolation curves. Thus, Mokhtarian and Mackworth [21] propose to filter the curve with a 1D 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 and Dinstein [2], 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 and Srinath [19] calculate 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 3×3 neighbourhood. Arrebola et al. [5] 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 with 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 are missed. 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 [7]. 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 [26].

Both approaches have been used to calculate the curvature function associated with a laser scan: the iterative curvature scale space (CSS) was used by Madhavan and Durrant-Whyte [20] 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 with the original shape can be identified [21]. On the other hand, the adaptive curvature function was employed by Núñez et al. [24] to extract corners, line and curve segments from the laser scan data. This curvature function is based on a modified version of the angle-based curvature estimator proposed in [26].

In this work, we use the adaptive curvature function to estimate the curvature at each range reading. Then, this information is employed to segment the laser scan into clusters of homogeneous curvature. The process to achieve this segmentation task [24] is briefly described in this paper, and consists of the following steps:

(1) Calculation of the maximum length of laser scan presenting no discontinuities on the right and left sides of the working range reading i: $K_f[i]$ and $K_b[i]$, respectively. $K_f[i]$ is calculated by comparing the Euclidean distance from range reading i to its $K_f[i]$th neighbour $(d(i, i + K_f[i]))$ to the length of the laser scan between both range readings $(l(i, i + K_f[i]))$ which is defined as

$$l(i, i + K_f[i]) = \sum_{j=i}^{K_f[i]-1} d(j, j + 1). \tag{4}$$

Both distances tend to be equal in absence the of corners, even if laser scans are noisy. Otherwise, the Euclidean distance is quite shorter than the scan length. Thus, $K_f[i]$

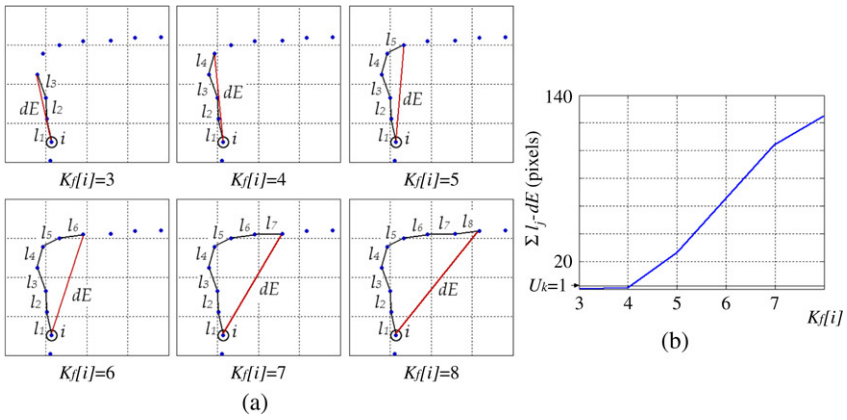

Fig. 4. Calculation of the maximum length of laser scan presenting no discontinuities on the right side of range reading i ($K_f[i]$): (a) Euclidean distance from range reading i to range reading $i + K_f[i]$ (dE) and length of the laser scan between both range readings ($l = \sum l_j$); and (b) $K_f[i]$ selection as the largest value that satisfies $\sum l_j - dE < U_k$ (in this case, $K_f[i] = 4$).

is the largest value that satisfies

$$l(i, i + K_f[i]) - d(i, i + K_f[i]) < U_k \tag{5}$$

U_k being a constant value that depends on the noise level tolerated by the detector. Fig. 4 shows the process to extract one $K_f[i]$ value. $K_b[i]$ is also set according to Eq. (5), but using $i - K_b[i]$ instead of $i + K_f[i]$. The correct selection of the U_k value is very important. Thus, if the value of U_k is large, $K_f[i]$ and $K_b[i]$ tend to be large and some corners may be missed and, if it is small, $K_f[i]$ and $K_b[i]$ are always very small and the resulting function is noisy. Section 5.2 shows the process employed to fix a suitable U_k.

Fig. 5(c) presents an example of the $K_f[i]$ and $K_b[i]$ values associated with the range readings in Fig. 5(a). It can be noted that the $K_f[i]$ and $K_b[i]$ values associated with range readings i located near to a corner are reduced in order to accommodate them to the laser scan contour. In this example, values for $K_f[i]$ and $K_b[i]$ have been confined to the interval $[3 \ldots 7]$.

(2) Calculation of the local vectors $\vec{f}_i$ and $\vec{b}_i$ associated with each range reading i. These vectors present the variation in the x- and y-axis between range readings i and $i + K_f[i]$, and between i and $i - K_b[i]$. If (x_i, y_i) are the Cartesian coordinates of the range reading i, the local vectors associated with i are defined as

$$\begin{aligned} \vec{f}_i &= (x_{i+K_f[i]} - x_i, y_{i+K_f[i]} - y_i) = (f_{x_i}, f_{y_i}) \\ \vec{b}_i &= (x_{i-K_b[i]} - x_i, y_{i-K_b[i]} - y_i) = (b_{x_i}, b_{y_i}). \end{aligned} \tag{6}$$

(3) Calculation of the angle associated with each range reading of the laser scan. According to the works of [27], the angle at range reading i can be estimated by using the equation:

$$\kappa_i = \arccos\left(\frac{\vec{f}_i \cdot \vec{b}_i}{|\vec{f}_i| \cdot |\vec{b}_i|}\right). \tag{7}$$

(4) Detection of line segments over κ_i. Line segments result from the scan of planar surfaces. Therefore, they are those set of consecutive range readings which are under a minimum angle ($\kappa_{\min}$).

(5) Detection of curve segments over κ_i. Curve segments result from the scan of curve surfaces. Contrary to the curvature values associated with a line segment, it can be appreciated that the curvature function associated with a curve segment presents consecutive local peaks whose absolute height can be greater than $\kappa_{\min}$. Some of them could be wrongly considered as corners. To avoid this error, Núñez et al. [24] associate a cornerity index to each set of consecutive range readings whose κ_i values are over $\kappa_{\min}$ or under $-\kappa_{\min}$. Thus, curve segments are those sets of consecutive range readings which do not define a line segment and have a cornerity index greater than a given threshold U_c [24].

Fig. 6 shows the final segmentation of a laser scan between two consecutive breakpoints. It can be noted that the curvature function around corners is not well defined. In this case, three regions have been detected (two line segments and one curve segment). It can also be appreciated that usually a real curve segment will be represented in the curvature function as a set of consecutive local peaks of similar curvature. These consecutive curve segments are grouped if they present similar curvature values. Line and curve segments are the input of the Landmark Extraction and Characterisation stage, which is described in the next chapter.

4. Natural landmark extraction and characterisation

As can be appreciated from Fig. 6, the adaptive curvature function can directly provide three different natural landmarks: line segments, corners and curve segments [24]. However, in order to include these items as landmarks in an EKF-based SLAM algorithm [33], it is necessary to characterise them by a set of invariant parameters and moreover, to estimate their uncertainties. This is typically achieved by fitting parametric curves to measurement data associated with each line or curve segment and evaluating the uncertainty associated with the measured data. Thus, line and curve segments can be used as stable landmarks. In this work, this is achieved using models that minimize the orthogonal distance from coordinate data to the line or curve segment. It is assumed that each range

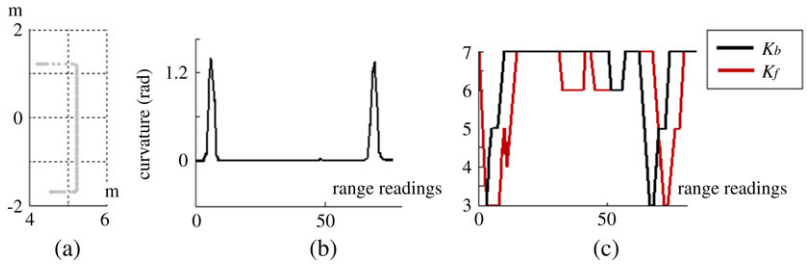

Fig. 5. (a) Segment of a laser scan; (b) curvature function associated with (a); (c) K_f and K_b values associated with the range readings of the laser scan in (a).

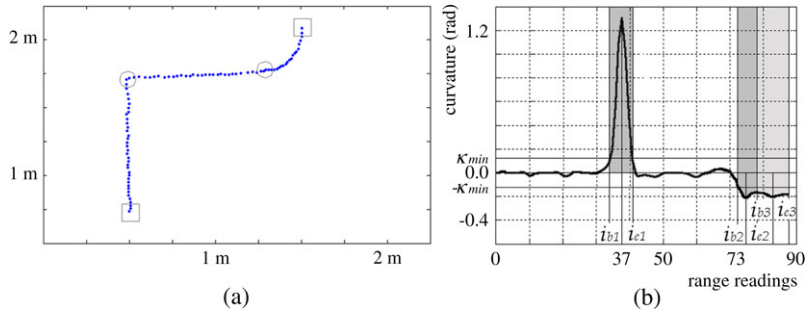

Fig. 6. (a) Segment of a laser scan; and (b) curvature function associated with (a). The figure shows the i_b and i_e values that bound the set of curvature values associated with each local peak.

reading is independently affected by Gaussian noise in both coordinates, range r_l and bearing ϕ_l. Finally, other types of landmarks are extracted and characterised as corners or edges. In this section, the process to describe each type of landmark is presented.

4.1. Line segments

In order to provide precise feature estimation it is essential to represent uncertainties and to propagate them from single range reading measurements to all stages involved in the feature estimation process. To achieve this, feature extraction with a Kalman filter has become a widely used approach [6,36]. In this approach, a measurement prediction based on a feature model is matched with its corresponding observation in order to compute the best estimate of the model parameters. However, the main drawback of this approach is the initialization of the filter when starting a new feature. It requires a first estimate of the feature position which has to be found without the Kalman filter. This regression problem becomes nonlinear in the parameters when geometrical meaningful errors have to be minimized [3].

There are several approaches for line fitting. Thus, the parameters of a straight-line in slope-intercept form can be determined using the equations for linear regression [35]. Then, the resulting line can be converted into the normal form representation

$$x\cos\theta + y\sin\theta = d \tag{8}$$

θ being the angle between the x-axis and the normal of the line and d the perpendicular distance of the line to the origin. Under the assumption of error free laser bearings, the covariance of the angle and distance estimate of the line can be derived. However, the problem of fitting a set of n points in Cartesian coordinates to a straight-line model using linear regression is based on the assumption that the uncertainty σ_i associated with each y_i is known and x_i values are known exactly. In our case, the points being processed in Cartesian coordinates are the result of a nonlinear transformation of points from polar coordinates:

$$x_i = r_i\cos\phi_i \qquad y_i = r_i\sin\phi_i. \tag{9}$$

This makes errors in both Cartesian coordinates correlated [13]. Effectively, if the errors in range and bearing are assumed to be independent with zero mean and standard deviations σ_r and σ_ϕ, respectively, then the covariance matrix associated with a range reading i in Cartesian coordinates can be approximated with a first-order Taylor expansion as

$$\begin{aligned} C_{xyi} &= J_r \begin{bmatrix} \sigma_\phi^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} J_r^{\mathrm{T}} \\ &= \begin{bmatrix} \sigma_r^2 s^2 + r_i^2\sigma_\phi^2 c^2 & \sigma_r^2 sc - r_i^2\sigma_\phi^2 cs \\ \sigma_r^2 sc - r_i^2\sigma_\phi^2 cs & \sigma_r^2 c^2 + r_i^2\sigma_\phi^2 s^2 \end{bmatrix} \end{aligned} \tag{10}$$

where J_r is the Jacobian of polar to Cartesian coordinates transformation, and c and s are $\cos\phi_i$ and $\sin\phi_i$, respectively. It is shown the existence of non-zero terms associated with σ_{xi}^2, σ_{yi}^2 and σ_{xyi}.

Therefore, a better approach for line fitting is to minimize the sum of square perpendicular distances of range readings to lines. This yields a nonlinear regression problem which can be solved for polar coordinates [3]. The line in the laser range finder's polar coordinate system is represented as

$$r = \frac{d}{\cos(\theta - \phi)} \tag{11}$$

where θ and d are the line parameters (Eq. (8)). Then, the orthogonal distance d_i of a range reading, $(r,\phi)_i$, to this line

is

$$d_i = r_i \cos(\theta - \phi_i) - d. \tag{12}$$

This distance d_i represents the error associated with the line at this point. Then, the sum of squared errors can be defined as

$$S_l(b) = \sum_{i=1}^{n} d_i^2 = \sum_{i=1}^{n} (r_i \cos(\theta - \phi_i) - d)^2 \tag{13}$$

n being the number of range readings that belong to the line segment and $b = (\theta d)^{\mathrm{T}}$ the parameter vector. Arras and Siegwart [3] propose to weight each single point by a different value w_i that depends on the variance modelling the uncertainty in radial and angular direction. This produces a weighted sum of squared errors. In our case, uncertainties in range and bearing are the same for every range reading, so the weights for each point in polar coordinates are also equal. Therefore, we have not employed these weights.

The model parameters of the line (θ, d) can be obtained by solving the nonlinear equation system to minimize (13)

$$\frac{\partial S_l(b)}{\partial \theta} = 0 \qquad \frac{\partial S_l(b)}{\partial d} = 0 \tag{14}$$

whose solution is [3]

$$\theta = \frac{1}{2} \arctan \left(\frac{\sum_i r_i^2 \sin 2\phi_i - \frac{2}{n} \sum_i \sum_j r_i r_j \cos\phi_i \sin\phi_j}{\sum_i r_i^2 \cos 2\phi_i - \frac{1}{n} \sum_i \sum_j r_i r_j \cos(\phi_i + \phi_j)} \right) \tag{15}$$

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

For computation reasons, in [3] the Cartesian form of (15) is suggested which presents a lower computational complexity

$$\theta = \frac{1}{2} \arctan \left(\frac{-2 \sum_i (\bar{y} - y_i)(\bar{x} - x_i)}{\sum_i [(\bar{y} - y_i)^2 - (\bar{x} - x_i)^2]} \right) = \frac{1}{2} \arctan \frac{N}{D} \tag{16}$$

$$d = \bar{x} \cos\theta + \bar{y} \sin\theta$$

where $\bar{x} = \sum r_i \cos\phi_i / n$ and $\bar{y} = \sum r_i \sin\phi_i / n$.

Finally, assuming that the individual measurements are independent, the covariance matrix of the estimated line parameters (θ, d) can be calculated as [13]

$$C_{\theta,d} = \sum_i^n J_i C_{xyi} J_i^{\mathrm{T}} = \sum_i^n \begin{bmatrix} \partial\theta/\partial x_i & \partial\theta/\partial y_i \\ \partial d/\partial x_i & \partial d/\partial y_i \end{bmatrix} C_{xyi} \begin{bmatrix} \partial\theta/\partial x_i & \partial d/\partial x_i \\ \partial\theta/\partial y_i & \partial d/\partial y_i \end{bmatrix} \tag{17}$$

where the terms $\partial\theta/\partial x_i$, $\partial\theta/\partial y_i$, $\partial d/\partial x_i$ and $\partial d/\partial y_i$ are obtained as follows

$$\begin{aligned} \frac{\partial\theta}{\partial x_i} &= \frac{(\bar{y} - y_i)D + (\bar{x} - x_i)N}{N^2 + D^2} \\ \frac{\partial\theta}{\partial y_i} &= \frac{(\bar{x} - x_i)D + (\bar{y} - y_i)N}{N^2 + D^2} \\ \frac{\partial d}{\partial x_i} &= \frac{1}{n}\cos\theta \\ &\quad + (\bar{y}\cos\theta - \bar{x}\sin\theta)\frac{(\bar{y} - y_i)D + (\bar{x} - x_i)N}{N^2 + D^2} \\ \frac{\partial d}{\partial y_i} &= \frac{1}{n}\sin\theta + (\bar{y}\cos\theta - \bar{x}\sin\theta)\frac{(\bar{x} - x_i)D + (\bar{y} - y_i)N}{N^2 + D^2} \end{aligned} \tag{18}$$

N and D being the numerator and denominator of the expression of θ (16).

Fig. 7 presents three different laser scans and the detected landmarks. Figs. 7(d)–(f) show the line segments extracted using the described approach and corresponding to laser scans in Figs. 7(a)–(c), respectively. The end-points of each line segment are determined by the intersection between this line and the two lines which are perpendiculars to it and pass through the first and last range readings (see Appendix A). Fig. 8 shows the calculation of one of the end-points of the line (θ_1, d_1).

4.2. Curve segments

A curve segment of constant curvature can be considered as an arc of a circle which is basically described by its center of curvature (x_c, y_c) and its radius ρ. Circle fitting problem estimates these parameters finding the vector $b = (x_c, y_c, \rho)$ that minimizes

$$S_c(b) = \sum_{i=1}^{n} [(x_i - x_c)^2 + (y_i - y_c)^2 - \rho^2]^2 \tag{19}$$

where $\{(x, y)\}_{i=1\ldots n}$ is the set of range readings that defines the curve segment in Cartesian coordinates. However, the covariance matrix associated with each range reading in Cartesian coordinates is different (see Eq. (10)) and then, each term in Eq. (19) must be weighted by a value which will take into account the measurement uncertainty. In our particular case, this can be avoided if we work in polar coordinates, because in this coordinate system, the covariance matrix is the same for each reading (Section 4.1). Therefore, our aim is to find the circle $(x_i - x_c)^2 + (y_i - y_c)^2 - \rho^2 = 0$ where $x = r\cos\phi$, $y = r\sin\phi$, $x_c = r_c\cos\phi_c$ and $y_c = r_c\sin\phi_c$, yielding

$$r^2 + r_c^2 - 2rr_c\cos(\phi - \phi_c) - \rho^2 = 0. \tag{20}$$

To minimize $S_c(b) = S_c(r_c, \phi_c, \rho)$, finding the parameter vector b, we use the Levenberg–Marquardt algorithm [22]. This algorithm approximates S_c as a linear function of b, $\hat{S}_c$:

$$S_c(b) \approx \hat{S_c}(b) = \sum (d_i(b_k) + \nabla d_i(b_k) \cdot b)^2 \tag{21}$$

where $d_i(b) = r_i^2 + r_c^2 - 2r_i r_c \cos(\phi_i - \phi_c) - \rho^2$ and $\nabla d_i(b)$ is the gradient of $d_i(b)$. This estimation is valid within a

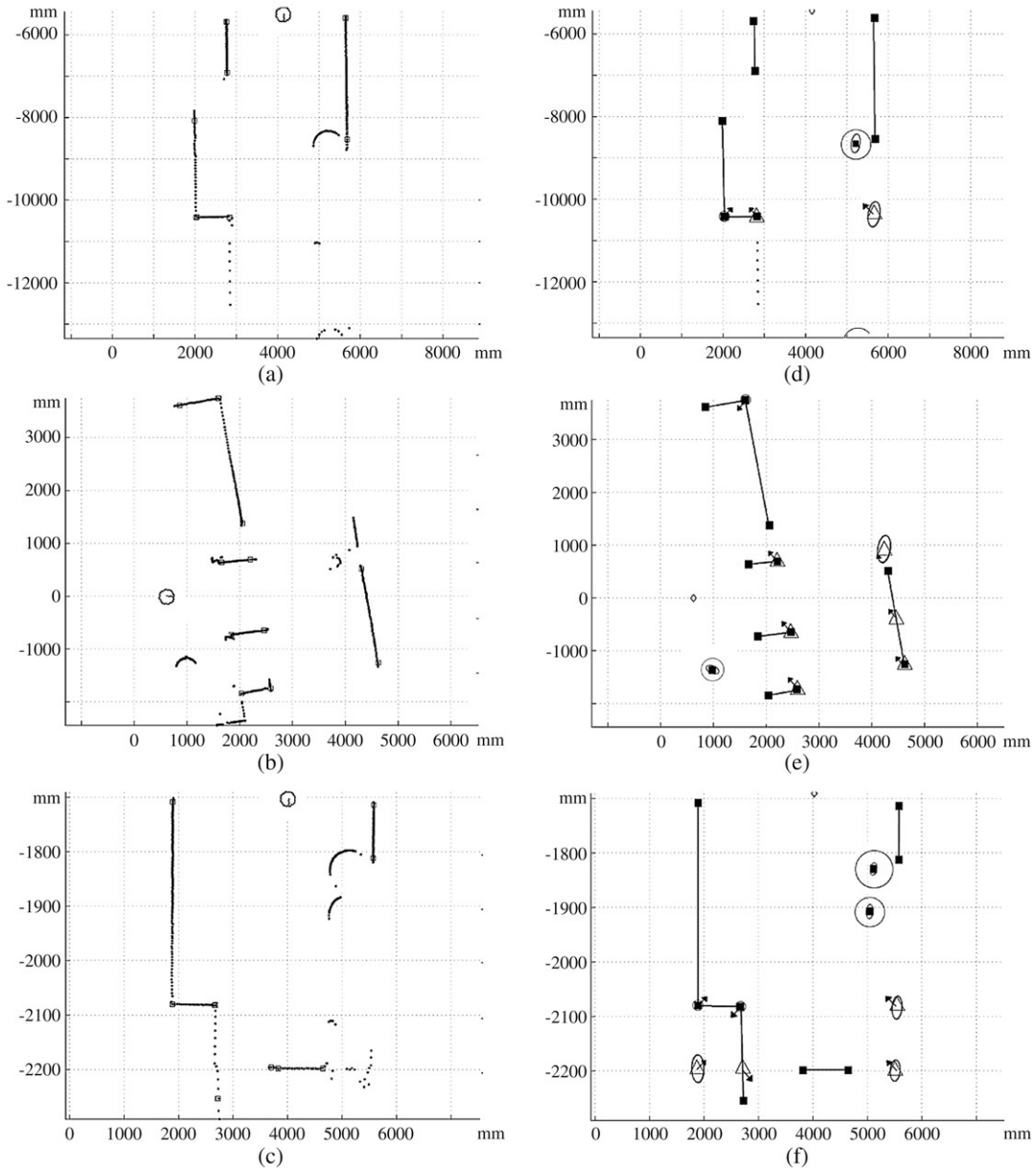

Fig. 7. (a)–(c) Three laser scans; and (d)–(f) segmentation and landmark detection associated with (a)–(c) (□—line segments end-points, o—real corners, △—virtual corners, →—corner orientations). Circles are also represented in the figure (center and circumference).

certain trust region radius. The algorithm begins using an initial parameter vector b_{cur}. Then, the derivation considers how to minimize $\hat{S}_c(p)$. A search direction is obtained based on the linear function of b, and a search is made in that direction within the limits of the trust region radius for a b_{new} such that $S_c(b_{\text{new}}) < S_c(b_{\text{cur}})$. When this b_{new} is found, it becomes the new b_{cur} for another iteration of the above process. At each iteration, the solution can be expressed as

$$\nabla b = b_{\text{new}} - b_{\text{cur}} = -(J_{\text{cur}}^{\text{T}} J_{\text{cur}} + \lambda D^{\text{T}} D)^{-1} J_{\text{cur}}^{\text{T}} d(b_{\text{cur}}) \quad (22)$$

J_{cur} being the Jacobian matrix of the algorithm having $\nabla d_i(b_{\text{cur}})$ as its ith row, D is a weighting matrix, $d(b_{\text{cur}})$ is the vector of residuals $d_i(b_{\text{cur}})$ and λ is a non-negative variable which can be considered the Lagrange multiplier for the constraint that each search is limited to the trust region radius.

The Levenberg–Marquardt algorithm consists of the following steps [29]:

- Set $\lambda = 1$; $k_1 = 0$
- Repeat
  - Set $k_2 = 0$; $k_1 = k_1 + 1$
  - $\lambda = 0.04 \cdot \lambda$
  - $b_{\text{cur}} = \frac{b_{\text{cur}}}{|b_{\text{cur}}|}$
  - Set $U = J_{\text{cur}}^{\text{T}} J_{\text{cur}}$, $v = J_{\text{cur}}^{\text{T}} d(b_{\text{cur}})$; $S_c(b_{\text{cur}}) = \sum (d_i(b_{\text{cur}}))^2$
  - Repeat
    $k_2 = k_2 + 1$
    $\lambda = 10 \cdot \lambda$

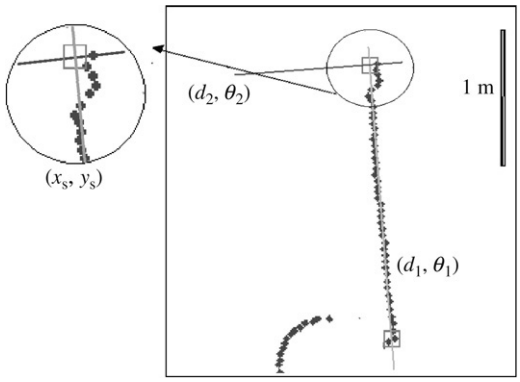

Fig. 8. End-points of line segment are calculated as the intersection between two perpendicular lines.

Set $H = U + \lambda(I + \text{diag}(u_{11}, u_{22} \ldots u_{nn}))$
Solve the system $Hx = -v$
Set $b_{\text{new}} = b_{\text{cur}} + x$; $J_{\text{new}} = \sum(d_i(b_{\text{new}}))^2$
If converged, set $b_{\text{cur}} = \frac{b_{\text{new}}}{|b_{\text{new}}|}$; return b_{cur}
· Until $S_c(b_{\text{new}}) < S_c(b_{\text{cur}})$ or $k_2 >$ NMAXITER
· If $S_c(b_{\text{new}}) < S_c(b_{\text{cur}})$ then $b_{\text{cur}} = b_{\text{new}}$
• Until $k_1 >$ NMAXITER

This algorithm uses a weighting matrix defined so that $D^{\text{T}}D$ is the identity matrix plus the diagonal of $J_{\text{cur}}^{\text{T}}J_{\text{cur}}$ [22]. The system $Hx = -v$ can be reliably solved using the Cholesky decomposition [29].

As it was pointed out above, a starting guess for parameters is required. To obtain this parameter vector b_0, we use the equation of a circle passing through three given points, (x_1, y_1), (x_2, y_2) and (x_3, y_3), (Appendix B). These three points correspond respectively to three different range readings belonging to the previously extracted curve segment. Finally, an estimation of the curve segment uncertainty, represented as $C_{(r_c,\phi_c,\rho)}$, can be derived from this same expressions (see Appendix C).

Figs. 7(a)–(c) present different real laser scans containing columns and tree-like elements that are extracted and represented (center and circumference) in Figs. 7(d)–(f). Uncertainties associated with the center of the circles are also shown.

4.3. Real corners

Corners are due to the change of surface being scanned or due to the change in the orientation of the scanned surface. Thus, they are not associated with laser scan discontinuities. In order to extract stable corners, the iterative curvature scale space (CSS) [20] or the adaptive curvature function [24] has been used. Both approaches locate the corner as one of the range readings of the laser scan and characterises it by its position in the Cartesian coordinate system. However, as it is illustrated in Fig. 9, the corner is not always located in one of the scan range readings. Failing to identify the correct corner point in the data can lead to large errors especially when corner the is distant from the robot.

Other option is to extract the corner taking into account the two lines associated with it. Thus, corner can be detected as the

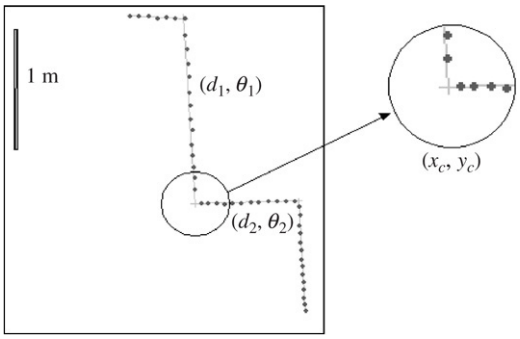

Fig. 9. A real corner is not usually located at one of the laser range readings (they are marked as black dots over the detected line segments).

furthest point from a line defined by the two non-touching end-points of the lines or by finding that point in the neighbourhood of the initial corner point, which gives the minimum sum of error variances of both lines [13]. In our case, the existence of a corner can be determined from the curvature function but its characterisation (estimation of the mean pose and uncertainty measurement) is conducted using the two lines which generate the corner. Therefore a corner will be always defined as the intersection of two lines, i.e. corners defined as the intersection of a curve and a line or of two curves will be not taken into account.

Although the corner value is a single curvature point, it is not defined in the curvature function as a Dirac delta function. Thus, the corner is always defined by a value associated with a local peak of the curvature function, and a region bounded by two range readings, i_b and i_e. Therefore, it can be characterised by the cornerity index c_i [24]. Taken this into account, corners are those range readings which do not belong to any line or curve segments and satisfy the following conditions: (i) they are local peaks of the curvature function, i.e. their $|\kappa_i|$ values are over the minimum angle required to be considered a corner instead of a spurious peak due to remaining noise ($\kappa_{\min}$); (ii) they are located between two segments which have been marked as line segments (these two segments determine the region of the corner, (i_b, i_e)); and (iii) their cornerity indexes are less than U_c. These constraints can be quickly applied if the curvature function has been previously obtained.

Once a corner is detected, its position (x_c, y_c) is estimated as the intersection of the two lines which generate it (see Appendix A). The corner orientation α_c can be also calculated as the bisector of the angle defined by these two lines [32]. Finally, the covariance of the estimated corner parameters C_{x_c,y_c,α_c} can be computed depending on the noise in the line parameters (see Appendix D). Fig. 7 illustrates the corner detection results. Poses and uncertainties, associated with the three laser scans in Figs. 7(a)–(c) are shown in Figs. 7(d)–(f). Real corners are marked as 'o' over the laser scans.

4.4. Virtual corners

As it has been pointed out by Madhavan and Durrant-Whyte [20], one of the main problems of a localization algorithm which is only based on corner detection is that the set of detected natural landmarks at each time step can

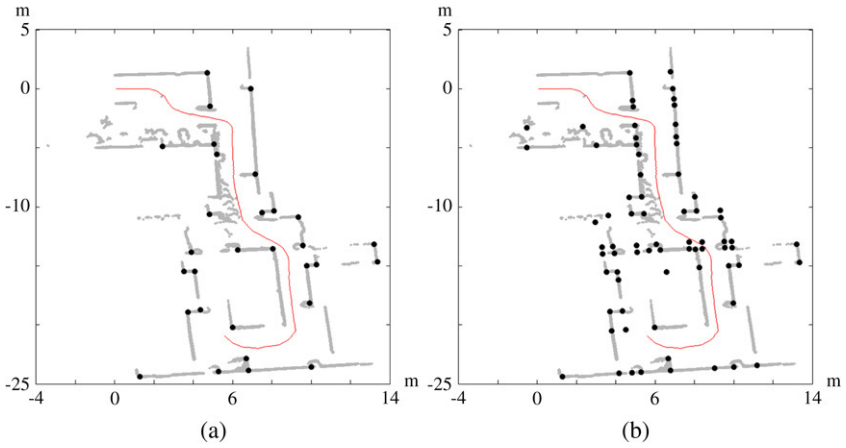

Fig. 10. Overlaid laser scans for one experiment: (a) Black dots show the real corner that were extracted using the proposed approach; and (b) black dots show the real and virtual corners that were extracted using the proposed approach. In both cases, the solid line represents the ground truth and the direction of travel is from top to bottom.

be very reduced, 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 attenuate this problem, we propose in this paper to include a new natural landmark which can be used in the same way as real corners: the virtual corner. Virtual corners are defined as the intersection of extended line segments which are not previously defined as real corners. Fig. 10 shows a set of overlaid laser scans. In Fig. 10(a), dots show the real corners that were extracted using the proposed approach, whereas Fig. 10(b) represent real and virtual corners. The number of landmarks in this experiment has been increased from 30 to 81.

The virtual corner proposed in this paper is related to the virtual edge anchor [34]. However, in our case, the virtual corner is related to the line segments previously extracted from the curvature function. The virtual edge anchor is found without explicit line extraction. Instead, "*geometrical statistics is used to determine the large objects positions and orientations*" [34]. Although the authors do not specify the used approach, they justify it because it offers higher robustness against partial occlusion and noise effects. In our approach, the robust detection of lines is directly related to the adaptive curvature estimation algorithm and the process used for line characterisation.

Finally, virtual corners can be characterised using the same process described for a real corner in Section 4.3. Figs. 7(d)–(f) show virtual corners (poses and uncertainties) associated with laser scans in Figs. 7(a)–(c). The error propagation due to the distances from the lines to the virtual corners can be appreciated in their uncertainty ellipses. Figs. 7(d)–(f) also present that virtual corners significantly increases the size of the extracted observation vector.

4.5. 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 with end-points of plane surfaces [36]. To satisfy this condition, the portion of the environment where the breakpoint is located must be a line segment and it must not be occluded by any other obstacle. This last condition is true if the breakpoint is nearer to the robot than the other breakpoint defined by the same large discontinuity (see Fig. 11). 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 not possible to know if they define the end-point of a surface.

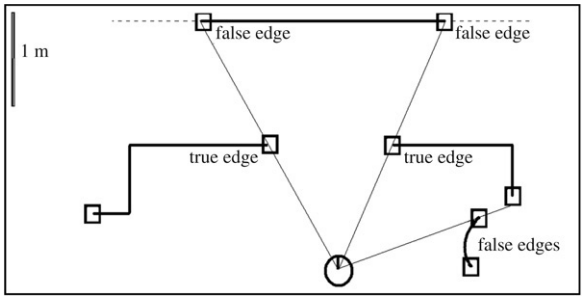

Fig. 11. An edge is defined as a breakpoint associated with the end-point of a plane surface which is not occluded by any other obstacle.

Edges are characterised by the Cartesian position (x, y) of the breakpoint and by the orientation of the plane surface described by the line segment, α. Therefore, the covariance of the estimated edge parameters (x_e, y_e, α_e) can be approximated as

$$C_{x_e y_e \alpha_e} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix} \tag{23}$$

where σ_α^2 is the orientation variance associated with the line segment.

5. Experimental results

5.1. Landmark detection results

As we commented above, the laser sensor used in the following experiments is a SICK LMS200 mounted on a

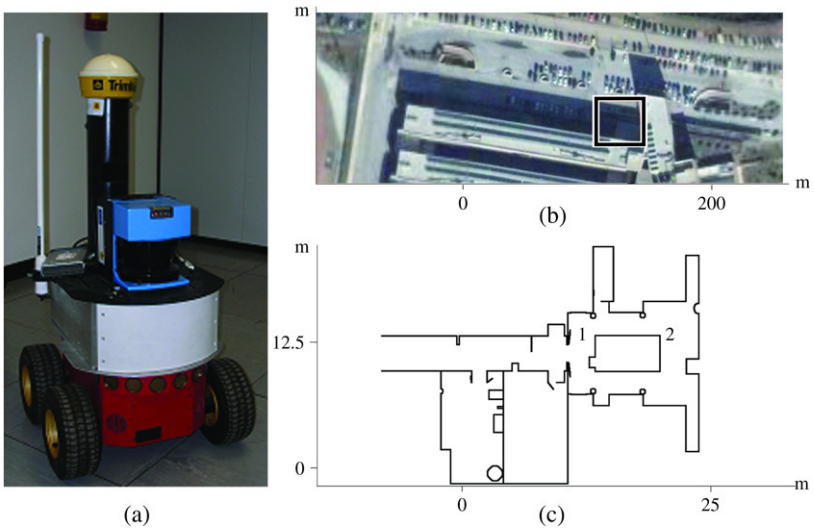

Fig. 12. (a) REX: a Pioneer 2AT robot from ActivMedia equipped with GPS and laser; (b) the first test area, Campus of Teatinos at Málaga University. Courtesy of Google Earth (http://earth.google.com); and (c) the second test area, an office-like environment sited at the Andalucía Technology Park (Málaga).

Pioneer 2AT robot from ActivMedia. 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. Fig. 12(a) shows the robot used in the experiments.

Fig. 12(b) shows the first test area within the campus of Teatinos at University of Málaga. This outdoor test area provides a semi-structured environment for the evaluation of the algorithm. Cars and people were present in this area while the experiment was carried out. Fig. 12(c) shows the second test area. In this case, the test area is an office-like environment which presents a higher density of detected landmarks. It must be noted that the set of threshold values used by the algorithm are the same for both scenarios.

Fig. 13 presents the results obtained from one experiment in the first test area, marked in Fig. 12(b). Fig. 13(b) presents the real data from one scan of the environment. It can be appreciated that in this type of environment the scan range readings are not equally distributed and moreover it is possible that some data segments were not able to be scanned. Fig. 13(c) shows that two columns, two cars and the building walls have been correctly observed. The line segments, center and circumference of curve segments and corners have been marked. Fig. 14 shows the landmark extraction results at two different robot poses marked in Fig. 12(c). They also present the effects of adaptive smoothing provided by the curvature estimation algorithm. Figs. 14(c) and 14(d) present two scan data collected in these poses. The laser scan range readings has been marked as dots over the real layout. The squares represent the start and end-points of each line segment. Detected corners and circles have been marked. Moreover uncertainty of corners and center of circles is presented as ellipses. Figs. 14(e) and 14(f) show the curvature functions associated with the laser scans in Figs. 14(c) and 14(d), respectively. The different segmented portions of the curvature functions are bounded by breakpoints or rupture points. All landmarks have been correctly detected.

Finally, Fig. 15 shows the analysis of several laser scans acquired by the robot of the same place from different points of view. It can be noted that all landmarks are correctly detected. In the figure, landmarks have been manually labelled according to the feature in the real scene they are related to. This figure shows the ability of the proposed approach to be included in a SLAM algorithm.

5.2. Estimation of parameters

One of the main drawbacks of the proposed method is the existence of a set of parameters to adjust. These parameters are:

- The systematic error $\epsilon_s(r_m)$ of the SICK LMS200 laser sensor.
- The parameters σ_r and λ used by the breakpoints detector.
- The threshold value which determines the noise level tolerated by the adaptive curvature detector, U_k.
- The minimum angle of a curvature value to be considered as a local peak of the curvature function, $\kappa_{\min}$.
- The minimum value of the cornerity index associated with a curve segment, U_c.

The process to approximate the systematic error of the SICK laser sensor by a polynomial has been briefly described and it is based on the previous work of Borges and Aldon [9]. Also based on this work, the σ_r and λ values have been fixed to 0.005 m and 10°, respectively.

The threshold values U_k and $\kappa_{\min}$ are used to eliminate spurious noise of the laser scan. In order to set them correctly, a set of real plane surfaces have been scanned at different distances from the robot. In these surfaces, the values must be fixed to not detect any local peak. This simple experiment has provided us an U_k value equal to 1.0 and a $\kappa_{\min}$ of 0.2 rad. These values have been used in all experiments described in this paper.

Finally, the threshold value U_c differentiates between corners and curve segments. In order to choose a correct U_c value, several cylindrical objects with different radius were

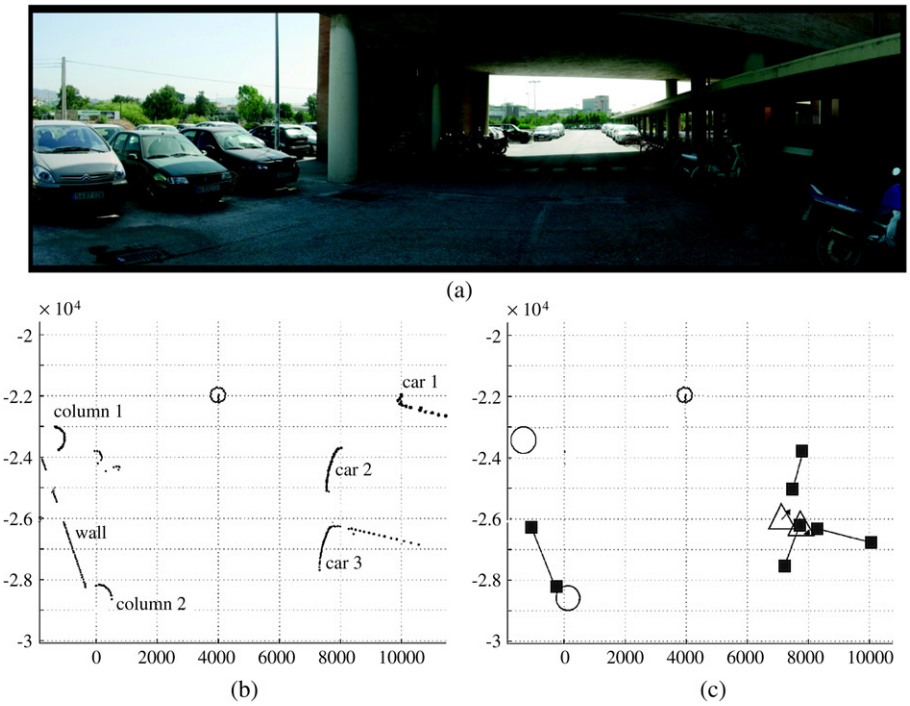

Fig. 13. (a) Real scene of the first test area; (b) Laser scan including cars, columns and buildings; and (c) Landmark detection (□—line segments end-points, △—virtual corners, →—corner orientations). Circles are also represented in the figure.

tested. In the segments of the curvature function associated with these objects, the proposed system should not detect any corner. From these experiments, a threshold value equal to 0.5 was selected.

6. A comparative study

In order to compare the proposed method to other approaches, we have implemented several segmentation algorithms which are commonly used for feature extraction in mobile robotics. Although these algorithms can also provide lines or corners, its use has been restricted to perform the segmentation task. In this way, they can be integrated as individual modules in our system and their results can be used for the same Landmark Extraction and Characterisation routines. This permits us to evaluate the speed, correctness and precision of these algorithms when they run into the same framework.

Particularly, we chose for comparison purposes methods described by Nguyen et al. [23], Zhang [35], Borges and Aldon [9], Madhavan and Durrant-Whyte [20] and Bandera et al. [8]. The first method is the split-and-merge (SM) algorithm. In a previous comparison of line extraction algorithms using 2D laser range finders, Nguyen et al. [23] conclude that this algorithm and the incremental one "*are preferred by their superior speed and correctness*" (see [23] for details). We have implemented the SM schema proposed in [23]. This method is a well-known algorithm for polygonal approximation in computer vision applications, which have problems in dealing with curve segments. In our case, every set of short line segments has been evaluated as a possible curve segment. In this way, circles can be detected and characterised. The SM algorithm will be applied to the laser scan segments obtained from the adaptive breakpoint detector [9]. Similar to the SM algorithm, the iterative-end-point-fit (IEPF) method provides a polygonal approximation to the laser scan. In this case, the version proposed by Zhang [35] have been implemented. The split-and-merge fuzzy (SMF) proposed by Borges and Aldon [9] uses fuzzy clustering in a split-and-merge framework without the need to guess the number of clusters. It has been also implemented. On the other hand, the approach proposed by Madhavan and Durrant-Whyte [20] uses point of maximum curvature, extracted from laser scan data, as point landmarks in an extended Kalman filter (EKF). A curvature scale space (CSS) algorithm is used to locate these points of maximum curvature. This method convolves the laser scan data with a Gaussian kernel and imparts smoothing at different levels of scale. In order to achieve a robust determination of dominant points, the algorithm detects them at the coarsest scale $\sigma_{\max}$ but it localizes the dominant point position at the finest scale $\sigma_{\min}$. In our case, the detected dominant points will be used to segment the laser scan between breakpoints, which are obtained using the adaptive breakpoint detector [9]. Experimental results will show the superior ability of this method to deal with curve segment with respect to the polygonal approximation approaches. Finally, the approach proposed by Bandera et al. [8] is a Hough-based method for extracting line segments from edge images that has been modified to deal with laser scans. Basically, this method uses a random window randomized Hough transform-(RWRHT) based approach to update an accurate Hough parameter space and a variable bandwidth mean shift algorithm to unsupervisedly cluster the items of this parameter space in a set of classes. Although the Hough transform can be also extended for fitting a circle, it has

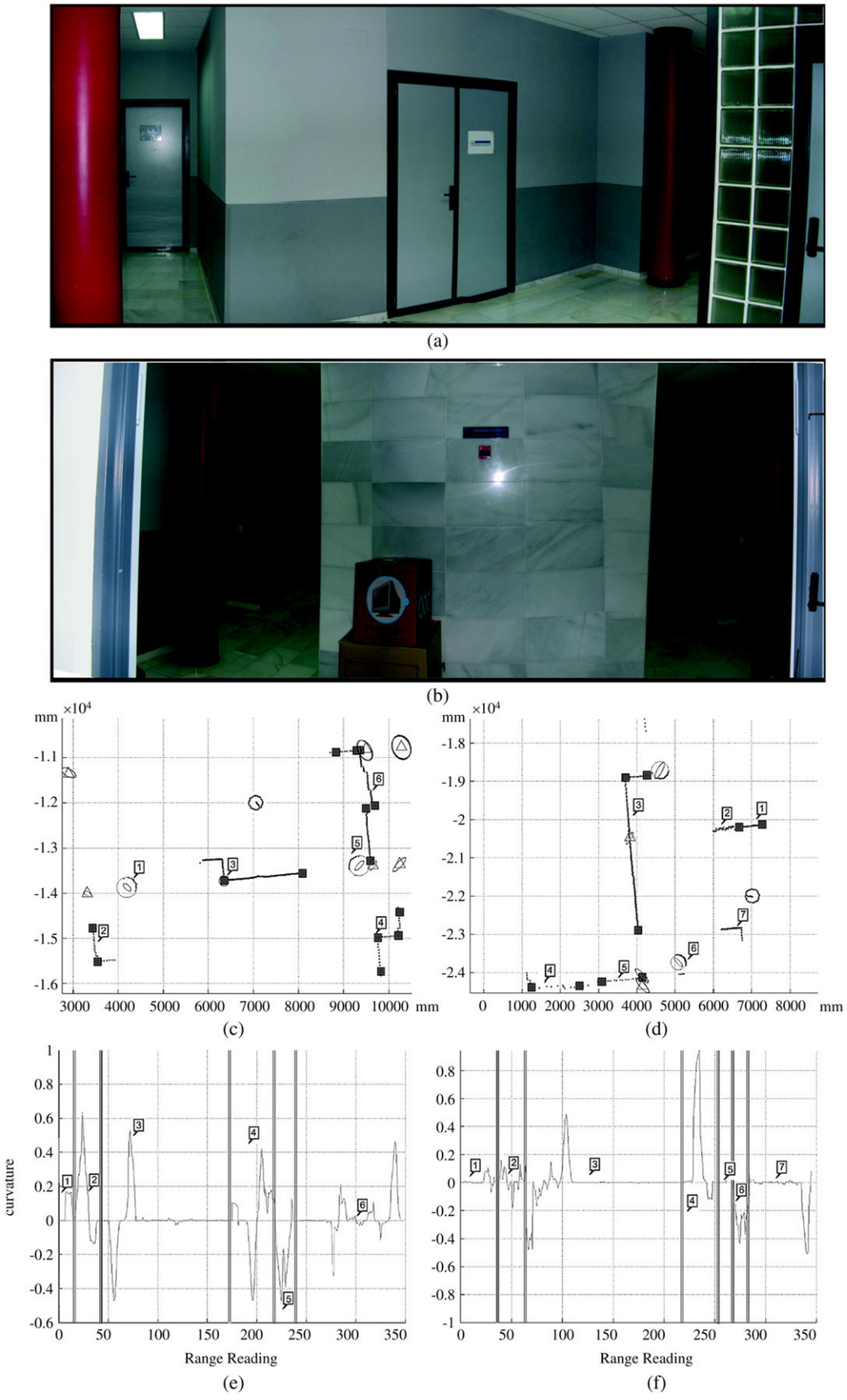

Fig. 14. (a), (b) Real scenes of the second test area, (c), (d) Landmark detection (□—line segments end-points, o—real corners, △—virtual corners). Circles are also represented in the figure; and (e), (f) curvature function associated with (a) and (b) respectively.

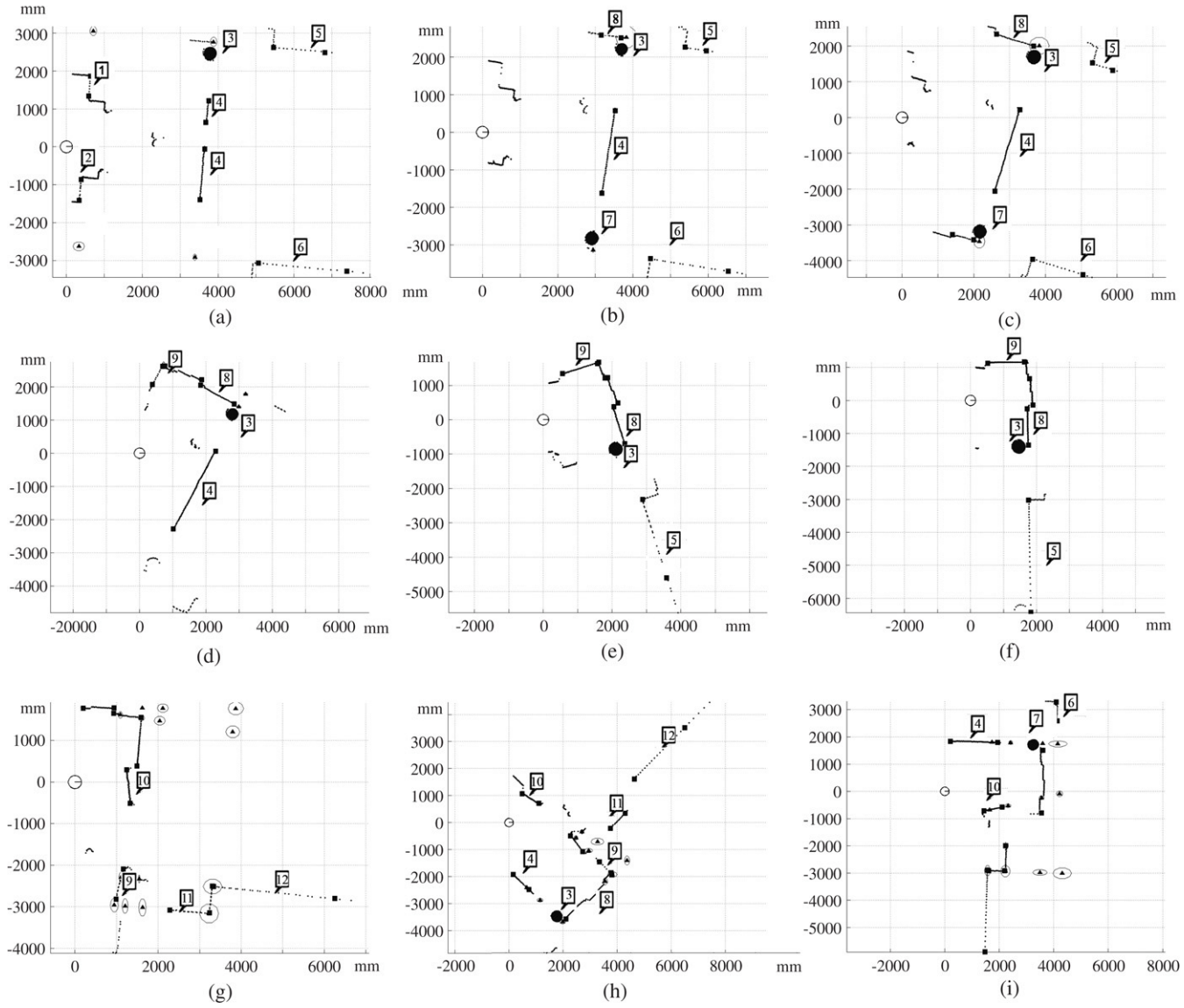

Fig. 15. (a), (f) Landmarks detected by the proposed approach. Laser scans have been acquired by the robot from different poses (see text).

not been implemented because a parameter space of three dimensions makes the problem more complex. Therefore, the Hough-based method will be only used to provide a segmentation of the laser scan.

To test the performance of the different approaches, a set of artificial maps have been created using the Mapper3 software from Activmedia Robotics. Laser scans have been obtained from these maps using MobileSim, a software for simulating mobile robots and their environments, for debugging and experimentation with ARIA-based control programs and other software. 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 $\sigma_\phi = 0.1°$. Each test scan consists of 360 range readings and it represents several line and curve segments. Thus, a total of 100 scans were generated for testing.

Algorithms have been programmed in C, and the benchmarks have been 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 ones. Each approach have been tested several times for the set of scans in order to obtain the best parameters. Segment pairs are initially matched using a χ^2-test with a matching valid gate value of 2.77. Then, extracted segments are matched to true segments using a nearest-neighbour algorithm. Experimental results are shown in Table 1. The correctness measures have been obtained as [23]

$$\begin{aligned} \text{TruePos} &= \frac{\text{NumberMatches}}{\text{NumberTrueSeg}} \\ \text{FalsePos} &= \frac{\text{NumberSegExAlg} - \text{NumberMatches}}{\text{NumberSegExAlg}} \end{aligned} \tag{24}$$

Table 1
Experimental results of the algorithms

Algorithm	SM	SMF	IEPF	CSS	HT	Proposed
Execution time (ms)	6.7	14.2	3.8	37.7	31.2	9.4
TruePos	0.76	0.79	0.77	0.92	0.71	0.91
FalsePos	0.19	0.19	0.21	0.02	0.23	0.02
$\sigma_{\Delta d}$ (mm)	14.5	11.7	15.1	10.5	10.3	10.6
$\sigma_{\Delta\theta}$ (°)	0.61	0.57	0.62	0.53	0.53	0.54
$\sigma_{\Delta x_c}$ (mm)	13.2	12.0	12.9	10.0	12.1	10.2
$\sigma_{\Delta y_c}$ (mm)	11.7	11.2	12.3	9.6	11.7	9.8
$\sigma_{\Delta\rho}$ (mm)	7.8	7.9	8.7	7.3	8.3	7.5

where NumberSegExAlg 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, only line and circle landmarks are taken into account. The following two sets of errors on line parameters are defined:

$$\begin{aligned} &\Delta d : \Delta d_i = |d_i - d_i^{\mathrm{T}}|, \quad i = 1 \ldots n \\ &\Delta\theta : \Delta\theta_i = |\theta_i - \theta_i^{\mathrm{T}}|, \quad i = 1 \ldots n \end{aligned} \tag{25}$$

where n is the number of matched pairs, d_i^{T} and θ_i^{T} are line parameters of a true line, and d_i and θ_i are line parameters of the corresponding matched line. It is assumed that error distributions are Gaussians. Similar sets of errors on circle parameters are defined.

As shown in the row 2 of the Table 1, the IEPF and SM algorithms are the fastest. The proposed method is significantly faster than the CSS and the HT ones, because its computational complexity is proportional to the laser scan length. In terms of correctness, the CSS and the proposed algorithms do not divide curve segments into short straight-line segments. Therefore, they have a high number of true positives and a few false positives. The rest of the methods usually associate one or two line segments to most of the curve segments present in the laser scans. This increases the number of false positives. To reduce these results, the minimum number of points per line segment can be increased. However, this leaves out short line segments, reducing the number of true positives. Finally, all algorithms have demonstrated its ability to provide precise lines. In this case, our results are very similar to the ones provided by Nguyen et al. [23] and Borges and Aldon [9]. With respect to circles, it can be noted that the CSS and the proposed method produce the best results. This is due to its ability to correctly detect curve segments.

7. Conclusions and future work

This paper presents an adaptive curvature estimation algorithm for segmenting range images obtained from a scanning laser rangefinder. With respect to other methods, which mainly try to split the laser scan into line segments, the main advantage of using curvature information is that the algorithm can provide line segments and curve segments. Besides, the main advantages of using an adaptive noise removal to estimate curvature are: (i) segmentation is performed at a wide range of scales for a constant set of parameters; and (ii) estimated curvature is better defined. Finally, the accuracy and robustness of the proposed method was demonstrated in several experiments while meeting real time requirements. Laser scan segments are used to obtain several types of landmarks. Line and curve segments permit to extract straight-line segments and circles. The curvature function also permits to detect the existence of real corners. However, these items will be characterised from the lines that define them. Finally, edges and virtual corners are also extracted from line segments. Covariance matrices associated with all landmarks are also provided.

Future work includes the development of an algorithm for robot localization based on the extracted features and to test it in dynamic environments. This algorithm must be capable to differentiate static and dynamic parts of the environment and therefore, to represent only these static parts on a map. The union of the static map and the moving objects could provide a complete description of the environment.

Appendix A. Finding the corner location as the intersection of two lines

Consider the two lines characterised by (ρ_1, d_1) and (ρ_2, d_2). The corner point (x_c, y_c) will be the intersection of these lines, so

$$\begin{aligned} &x_c \cos\theta_1 + y_c \sin\theta_1 - d_1 = 0 \\ &x_c \cos\theta_2 + y_c \sin\theta_2 - d_2 = 0. \end{aligned} \tag{A.1}$$

The first equation of (A.1) gives us an expression for x_c

$$x_c = \frac{d_1 - y_c \sin\theta_1}{\cos\theta_1}. \tag{A.2}$$

If we substitute this expression in the second equation of (A.1), we get

$$\frac{d_1 - y_c \sin\theta_1}{\cos\theta_1} \cos\theta_2 + y_c \sin\theta_2 - d_2 = 0 \tag{A.3}$$

$$d_1 \cos\theta_2 - y_c \sin\theta_1 \cos\theta_2 + y_c \cos\theta_1 \sin\theta_2 - d_2 \cos\theta_1 = 0 \tag{A.4}$$

$$y_c[\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2] = d_2 \cos\theta_1 - d_1 \cos\theta_2 \tag{A.5}$$

$$y_c = \frac{d_2 \cos\theta_1 - d_1 \cos\theta_2}{\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2} \tag{A.6}$$

which can be simplified to

$$y_c = \frac{d_2 \cos\theta_1 - d_1 \cos\theta_2}{\sin(\theta_2 - \theta_1)}. \tag{A.7}$$

Finally, we can substitute (A.7) in (A.2) to get

$$x_c = \frac{d_1 - \left(\frac{d_2 \cos\theta_1 - d_1 \cos\theta_2}{\sin\theta_2 \cos\theta_1 - \sin\theta_1 \cos\theta_2}\right)}{\cos\theta_1} \tag{A.8}$$

$$x_c = \frac{d_1(\sin\theta_2 \cos\theta_1 - \sin\theta_1 \cos\theta_2) - d_2 \cos\theta_1 \sin\theta_1 + d_1 \cos\theta_2 \sin\theta_1}{\cos\theta_1(\sin\theta_2 \cos\theta_1 - \sin\theta_1 \cos\theta_2)} \tag{A.9}$$

$$x_c = \frac{d_1 \sin\theta_2 - d_2 \sin\theta_1}{\sin(\theta_2 - \theta_1)}. \tag{A.10}$$

Appendix B. Finding the parameter vector of a circle segment using three points

Consider three points (x_1, y_1), (x_2, y_2) and (x_3, y_3) that belong to a circumference defined by $b = (x_c, y_c, \rho)$, where (x_c, y_c) is the center and ρ is the radius. In order to obtain b, solution of the problem, we can take into account that the distance between the three points and the circumference center is equal to the circumference radius, then

$$\begin{aligned}\rho^2 &= (x_1 - x_c)^2 + (y_1 - y_c)^2\\ \rho^2 &= (x_2 - x_c)^2 + (y_2 - y_c)^2\\ \rho^2 &= (x_3 - x_c)^2 + (y_3 - y_c)^2.\end{aligned} \tag{B.1}$$

The solution of the system defined by (B.1) is

$$\begin{aligned}y_c &= (a \cdot f - c \cdot d)/(b \cdot d - a \cdot e)\\ x_c &= (y_c \cdot b)/a + c/a\\ \rho &= \sqrt{((x_1 - x_c)^2 + (y_1 - y_c)^2)}\end{aligned} \tag{B.2}$$

where a, b, c, d, e and f are given by

$$\begin{aligned}a &= 2(x_2 - x_1)\\ b &= 2(y_2 - y_1)\\ c &= x_1^2 + y_1^2 - x_2^2 - y_2^2\\ d &= 2(x_3 - x_1)\\ e &= 2(y_3 - y_1)\\ f &= x_1^2 + y_1^2 - x_3^2 - y_3^2.\end{aligned} \tag{B.3}$$

Appendix C. Deriving the covariance matrix associated with a circle segment

A circle is characterised using its center and radius, (x_c, y_c, ρ). In this appendix we are interested in estimating the uncertainty associated with a circle segment. To calculate this uncertainty, denominated $C_{x_c,y_c,\rho}$, we define $\mathbf{b}$ as $[x_c y_c \rho]^{\mathrm{T}} = f(x_c, y_c, \rho)$. Then the first-order Taylor expansion of $\mathbf{b}$ is

$$\Delta\mathbf{b} = \nabla f(\mathbf{x}, \mathbf{y})\Delta[\mathbf{x}^{\mathrm{T}}\mathbf{y}^{\mathrm{T}}]^{\mathrm{T}} = J\Delta[\mathbf{x}^{\mathrm{T}}\mathbf{y}^{\mathrm{T}}]^{\mathrm{T}} \tag{C.1}$$

J being the Jacobian of $f(\mathbf{x}, \mathbf{y})$, whose elements are obtained by taking into account Eqs. (B.1) and (B.2). If we would calculate this Jacobian, we can approximate the covariance as

$$C_{(x_c,y_c,\rho)} = JC_{x_1y_1x_2y_2x_3y_3}J^{\mathrm{T}} \tag{C.2}$$

where $C_{x_1y_1x_2y_2x_3y_3}$ is known and defined as the matrix

$$C_{x_1y_1x_2y_2x_3y_3} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1y_1} & 0 & 0 & 0 & 0\\ \sigma_{x_1y_1} & \sigma_{y_1}^2 & 0 & 0 & 0 & 0\\ 0 & 0 & \sigma_{x_2}^2 & \sigma_{x_2y_2} & 0 & 0\\ 0 & 0 & \sigma_{x_2y_2} & \sigma_{y_2}^2 & 0 & 0\\ 0 & 0 & 0 & 0 & \sigma_{x_3}^2 & \sigma_{x_3y_3}\\ 0 & 0 & 0 & 0 & \sigma_{x_3y_3} & \sigma_{y_3}^2 \end{bmatrix}. \tag{C.3}$$

To obtain an approximate covariance and calculate the Jacobian, we can consider

$$\begin{aligned}y_c &= \frac{af - cd}{bd - ae} = \frac{N}{D}\\ R &= \rho^2\end{aligned} \tag{C.4}$$

where the constants a, b, c, d, e, and f are defined in (B.3). Then, the elements of the Jacobian can be calculated as

$$J_{1,1} = \frac{\partial x_c}{\partial x_1} = \frac{\partial y_c}{\partial x_1}\cdot\frac{b}{a} + \frac{2}{a^2}\cdot(by_c + x_1a + c) \tag{C.5}$$

$$J_{1,2} = \frac{\partial x_c}{\partial y_1} = \frac{\frac{\partial y_c}{\partial y_1}\cdot b + 2(y_1 - y_c)}{a} \tag{C.6}$$

$$J_{1,3} = \frac{\partial x_c}{\partial x_2} = \frac{\partial y_c}{\partial x_2}\cdot\frac{b}{a} - \frac{2}{a^2}\cdot(by_c + x_2a + c) \tag{C.7}$$

$$J_{1,4} = \frac{\partial x_c}{\partial y_2} = \frac{\frac{\partial y_c}{\partial y_2}\cdot b + 2(y_c - y_2)}{a} \tag{C.8}$$

$$J_{1,5} = \frac{\partial x_c}{\partial x_3} = \frac{\partial y_c}{\partial x_3}\cdot\frac{b}{a} \tag{C.9}$$

$$J_{1,6} = \frac{\partial x_c}{\partial y_3} = \frac{\frac{\partial y_c}{\partial y_3}\cdot b}{a} \tag{C.10}$$

$$\begin{aligned}J_{2,1} &= \frac{\partial y_c}{\partial x_1}\\ &= \frac{2(x_1\cdot(a-d) + c\cdot f)\cdot D - 2(e-b)\cdot N}{D^2}\end{aligned} \tag{C.11}$$

$$J_{2,2} = \frac{\partial y_c}{\partial y_1} = \frac{2y_1(a-d)\cdot D - 2(1-d)N}{D^2} \tag{C.12}$$

$$J_{2,3} = \frac{\partial y_c}{\partial x_2} = \frac{2x_2\cdot(f+d)\cdot D + 2eN}{D^2} \tag{C.13}$$

$$J_{2,4} = \frac{\partial y_c}{\partial y_2} = \frac{2y_2dD - 2dN}{D^2} \tag{C.14}$$

$$J_{2,5} = \frac{\partial y_c}{\partial x_3} = \frac{-2(ax_2 - c)\cdot D - 2bN}{D^2} \tag{C.15}$$

$$J_{2,6} = \frac{\partial y_c}{\partial y_3} = \frac{-2y_3aD + 2aN}{D^2} \tag{C.16}$$

$$\begin{aligned}J_{3,1} &= \frac{\partial\rho}{\partial x_1}\\ &= R^{-1/2}\left((x_1 - x_c)\left(1 - \frac{\partial x_c}{\partial x_1}\right) - (y_1 - y_c)\frac{\partial y_c}{\partial x_1}\right)\end{aligned} \tag{C.17}$$

$$\begin{aligned}J_{3,2} &= \frac{\partial\rho}{\partial y_1}\\ &= R^{-1/2}\left((y_1 - y_c)\left(1 - \frac{\partial y_c}{\partial y_1}\right) - (x_1 - x_c)\frac{\partial x_c}{\partial y_1}\right)\end{aligned} \tag{C.18}$$

$$\begin{aligned}J_{3,3} &= \frac{\partial\rho}{\partial x_2}\\ &= -R^{-1/2}\left((x_1 - x_c)\frac{\partial x_c}{\partial x_2} - (y_1 - y_c)\frac{\partial y_c}{\partial x_2}\right)\end{aligned} \tag{C.19}$$

$$J_{3,4} = \frac{\partial \rho}{\partial y_2} = -R^{-1/2}\left((y_1 - y_c)\frac{\partial y_c}{\partial y_2} + (x_1 - x_c)\frac{\partial x_c}{\partial y_2}\right) \quad \text{(C.20)}$$

$$J_{3,5} = \frac{\partial \rho}{\partial x_3} = -R^{-1/2}\left((x_1 - x_c)\frac{\partial x_c}{\partial x_3} + (y_1 - y_c)\frac{\partial y_c}{\partial x_3}\right) \quad \text{(C.21)}$$

$$J_{3,6} = \frac{\partial \rho}{\partial y_3} = -R^{-1/2}\left((y_1 - y_c)\frac{\partial y_c}{\partial y_3} + (x_1 - x_c)\frac{\partial x_c}{\partial y_3}\right). \quad \text{(C.22)}$$

Appendix D. Deriving the covariance matrix associated with a corner

To obtain an approximation of the uncertainty associated with a corner, represented as $C_{x_c y_c \alpha_c}$, we will operate in a similar way to Appendix C. In this case, we use $\mathbf{b}$ as $[x_c y_c \alpha_c]^{\mathrm{T}} = f(x_c, y_c, \alpha_c)$.

Therefore, after the first-order Taylor expansion, the covariance can be approximated as

$$C_{x_c y_c \alpha_c} = J C_{r_1 \alpha_1 r_2 \alpha_2} J^{\mathrm{T}} \quad \text{(D.1)}$$

$C_{r_1 \alpha_1 r_2 \alpha_2}$ being the covariance matrix associated with the lines which generate the corner. $C_{r_1 \alpha_1 r_2 \alpha_2}$ can be expressed as

$$C_{r_1 \alpha_1 r_2 \alpha_2} = \begin{bmatrix} \sigma_{r_1}^2 & \sigma_{r_1 \alpha_1} & 0 & 0 \\ \sigma_{r_1 \alpha_1} & \sigma_{\alpha_1}^2 & 0 & 0 \\ 0 & 0 & \sigma_{r_2}^2 & \sigma_{r_2 \alpha_2} \\ 0 & 0 & \sigma_{r_2 \alpha_2} & \sigma_{\alpha_2}^2 \end{bmatrix}. \quad \text{(D.2)}$$

Taking into account that $\alpha = (\alpha_1 - \alpha_2)/2$, the elements of the Jacobian J can be calculated as follows

$$J_{1,1} = \frac{\partial x_c}{\partial r_1} = \frac{\sin \alpha_2}{\sin(\alpha_2 - \alpha_1)} \quad \text{(D.3)}$$

$$J_{2,1} = \frac{\partial x_c}{\partial \alpha_1} = \frac{-r_2 \cos \alpha_1 \sin(\alpha_2 - \alpha_1) + \cos(\alpha_2 - \alpha_1)[r_1 \sin \alpha_2 - r_2 \sin \alpha_1]}{\sin^2(\alpha_2 - \alpha_1)} \quad \text{(D.4)}$$

$$J_{3,1} = \frac{\partial x_c}{\partial r_2} = \frac{-\sin \alpha_1}{\sin(\alpha_2 - \alpha_1)} \quad \text{(D.5)}$$

$$J_{4,1} = \frac{\partial x_c}{\partial \alpha_2} = \frac{r_1 \cos \alpha_2 \sin(\alpha_2 - \alpha_1) - \cos(\alpha_2 - \alpha_1)[r_1 \sin \alpha_2 - r_2 \sin \alpha_1]}{\sin^2(\alpha_2 - \alpha_1)} \quad \text{(D.6)}$$

$$J_{1,2} = \frac{\partial y_c}{\partial r_1} = \frac{-\cos \alpha_2}{\sin(\alpha_2 - \alpha_1)} \quad \text{(D.7)}$$

$$J_{2,2} = \frac{\partial y_c}{\partial \alpha_1} = \frac{-r_2 \sin \alpha_1 \sin(\alpha_2 - \alpha_1) + \cos(\alpha_2 - \alpha_1)[r_2 \cos \alpha_1 - r_1 \cos \alpha_2]}{\sin^2(\alpha_2 - \alpha_1)} \quad \text{(D.8)}$$

$$J_{3,2} = \frac{\partial y_c}{\partial r_2} = \frac{\cos \alpha_1}{\sin(\alpha_2 - \alpha_1)} \quad \text{(D.9)}$$

$$J_{4,2} = \frac{\partial y_c}{\partial \alpha_2} = \frac{r_1 \sin \alpha_2 \sin(\alpha_2 - \alpha_1) - \cos(\alpha_2 - \alpha_1)[r_2 \cos \alpha_1 - r_1 \cos \alpha_2]}{\sin^2(\alpha_2 - \alpha_1)} \quad \text{(D.10)}$$

$$J_{1,3} = \frac{\partial \alpha_c}{\partial r_1} = 0 \quad \text{(D.11)}$$

$$J_{2,3} = \frac{\partial \alpha_c}{\partial \alpha_1} = \frac{1}{2} \quad \text{(D.12)}$$

$$J_{3,3} = \frac{\partial \alpha_c}{\partial r_2} = 0 \quad \text{(D.13)}$$

$$J_{4,3} = \frac{\partial \alpha_c}{\partial \alpha_2} = -\frac{1}{2}. \quad \text{(D.14)}$$

References

[1] M.D. Adams, A. Kertens, Tracking naturally occurring indoor features in 2D and 3D with lidar range/amplitude data, International Journal of Robotics Research 17 (1998) 907–923.

[2] G. Agam, I. Dinstein, Geometric separation of partially overlapping nonrigid objects applied to automatic chromosome classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (19) (1997) 1212–1222.

[3] K.O. Arras, R. Siegwart, Feature extraction and scene interpretation for map-based navigation and map building, in: Proceedings of SPIE, in: Mobile Robotics XII, vol. 3210, 1997, p. 4253.

[4] K.O. Arras, N. Tomatis, B.T. Jensen, R. Siegwart, Multisensor on-the-fly localization: Precision and reliability for applications, Robotics and Autonomous Systems 34 (2001) 131–143.

[5] F. Arrebola, A. Bandera, P. Camacho, F. Sandoval, Corner detection by local histograms of the contour chain code, Electronics Letters 33 (21) (1997) 1769–1771.

[6] N. Ayache, O.D. Faugeras, Maintaining representation of the environment of a mobile robot, in: I.J. Cox, G.T. Wilfong (Eds.), Autonomous Robot Vehicles, Springer, 1990.

[7] A. Bandera, C. Urdiales, F. Arrebola, F. Sandoval, Corner detection by means of an adaptively estimated curvature function, Electronics Letters 36 (2) (2000) 124–126.

[8] A. Bandera, J.M. Pérez-Lorenzo, J.P. Bandera, F. Sandoval, Mean shift based clustering of Hough domain for fast line segment detection, Pattern Recognition Letters 27 (6) (2006) 578–586.

[9] G.A. Borges, M. Aldon, Line extraction in 2D range images for mobile robotics, Journal of Intelligent and Robotic Systems 40 (2004) 267–297.

[10] J.A. Castellanos, J.D. Tardós, Laser-based segmentation and localization for a mobile robot, in: M. Jamshidi, F. Pin, P. Dauchez (Eds.), Robotics and Manufacturing: Recent Trends in Research and Applications 6, ASME Press, 1996.

[11] J.A. Castellanos, J.M. Martínez, J. Neira, J.D. Tardós, Simultaneous map building and localization for mobile robots: A multisensor fusion approach, in: Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation, 1998, pp. 1244–1249.

[12] A.J. Davison, D.W. Murray, Simultaneous localisation and map-building using active vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 865–880.

[13] A. Diosi, L. Kleeman, 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.

[14] A. Elfes, Sonar-based real-world mapping and navigation, IEEE Journal of Robotics and Automation 3 (3) (1987) 249–265.

[15] L. Fontoura, R. Marcondes, Shape Analysis and Classification, CRC Press, 2001.

[16] J. Guivant, E. Nebot, S. Baiker, Localisation and map building using laser range sensors in outdoor applications, Journal of Robotic Systems 17 (10) (2000) 565–583.

[17] J. Kosecka, F. Li, X. Yang, Global localization and relative positioning based on scale-invariant keypoints, Robotics and Autonomous Systems 52 (1) (2005) 27–38.
[18] B.J. Kuipers, The spatial semantic hierarchy, Artificial Intelligence 119 (2000) 191–233.
[19] H. Liu, D. Srinath, Partial shape classification using contour matching in distance transformation, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (12) (1990) 1072–1079.
[20] R. Madhavan, H.F. Durrant-Whyte, Natural landmark-based autonomous vehicle navigation, Robotics and Autonomous Systems 46 (2004) 79–95.
[21] F. Mokhtarian, A. Mackworth, Scale-based description and recognition of planar curves and two-dimensional shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1) (1986) 34–43.
[22] J.C. Nash, Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation, Adam Hilger Ltd., 1979.
[23] V. Nguyen, A. Martinelli, N. Tomatis, R. Siegwart, A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'05, 2005, pp. 1929–1934.
[24] P. Núñez, R. Vázquez-Martín, J.C. del Toro, A. Bandera, F. Sandoval, Feature extraction from laser scan data based on curvature estimation for mobile robotics, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2006, pp. 1167–1172.
[25] N.E. Pears, Feature extraction and tracking for scanning range sensors, Robotics and Autonomous Systems 33 (2000) 43–58.
[26] P. Reche, C. Urdiales, A. Bandera, C. Trazegnies, F. Sandoval, Corner detection by means of contour local vectors, Electronics Letters 38 (14) (2002) 699–701.
[27] A. Rosenfeld, E. Johnston, Angle detection on digital curves, IEEE Transactions on Computers 22 (1973) 875–878.
[28] S. Roumeliotis, G.A. Bekey, SEGMENTS: A layered, dual-kalman filter algorithm for indoor feature extraction, in: Proc. of the 2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2000, pp. 454–461.
[29] C. Shakarji, Least-squares fitting algorithms of the NIST algorithm testing system, Journal of Research of the National Institute of Standards and Technology 103 (6) (1998) 633–641.
[30] J.D. Tardós, J. Neira, P.M. Newman, J.J. Leonard, Robust mapping and localization in indoor environments using sonar data, International Journal of Robotics Research (2002) 311–330.
[31] S. Thrun, An online mapping algorithm for teams of mobile robots, International Journal of Robotics Research 20 (5) (2001) 335–363.
[32] N. Tomatis, Hybrid, metric-topological, mobile robot navigation, Thèse n. 2444, Ecole Polytechnique fédérale de Lausanne (EPFL), Lausanne, 2001.
[33] R. Vázquez-Martín, P. Núñez, J.C. del Toro, A. Bandera, F. Sandoval, Adaptative observation covariance for EKF-SLAM in indoor environment using laser data, in: Proc. of the 13th IEEE Mediterranean Electrotechnical Conference, MELECON 2006, 2006.
[34] J. Weber, L. Franken, K. Jörg, E. Puttkamer, Reference scan matching for global self-localization, Robotics and Automation 40 (2002) 99–110.
[35] L. Zhang, B.K. Ghosh, Line segment based map building and localization using 2D laser rangefinder, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2000, pp. 2538–2543.
[36] S. Zhang, L. Xie, M.D. Adams, Feature extraction for outdoor mobile robot navigation based on a modified Gauss–Newton optimization approach, Robotics and Autonomous Systems 54 (2006) 277–287.

P. Núñez was born in Spain in 1978. He received the title of Telecommunication Engineering from the University of Malaga, Spain, in 2003. In 2007 he joined the University of Extremadura as Assistant Professor in the Department of Tecnología de los Computadores y Comunicaciones. He is currently a research associate and Ph.D. Student at the University of Malaga. Past stay was with the Institute of System and Robotics (ISR, University of Coimbra, Portugal). His research interests include mobile robot localization and landmarks extraction and characterisation.

R. Vázquez-Martín was born in Madrid, Spain, in 1975. He received the M.S. degree in mechanical engineering from the University of Málaga, Málaga, Spain, in 2002, majored in automation and electronics. After some years of working in companies related to industrial automation, in 2003 he returned to the University of Málaga to work as research assistant in the Electronic Technology Department. He is currently involved in his Ph.D., and his research interests include simultaneous localisation and map building, feature extraction and software engineering.

J.C. del Toro was born in Spain in 1976. He received his title of Specialistic Technician of Industrial Electronics in 1996, and his title of Telecommunication Technical Engineering from the Technical University School of Málaga, Spain, in 2002. In 2002 he joined the Department of Tecnología Electrónica of the University of Málaga where he works as a technician support to investigation. He is specializing in mid-range microcontroller systems. His research is focused on robotics and he is co-author of scientific journal papers and communications in International Conferences.

A. Bandera was born in Spain in 1971. He received his title of Telecommunication Engineering and Ph.D. degree from the University of Malaga, Spain, in 1995 and 2000, respectively. During 1997 he worked in a research project under a grant by the spanish CYCIT. Since 1998 he has worked as Assistant Professor and Lecturer successively in the Department of Tecnologia Electronica of the University of Malaga. His research is focused on robotics and artificial vision.

F. Sandoval was born in Spain in 1947. He received the title of Telecommunication Engineering and Ph.D. degree from the Technical University of Madrid, Spain, in 1972 and 1980, respectively. From 1972 to 1989 he was engaged in teaching and research in the fields of opto-electronics and integrated circuits in the Universidad Politécnica de Madrid (UPM) as an Assistant Professor and a Lecturer successively. In 1990 he joined the University of Málaga as Full Professor in the Department of Tecnología Electrónica. He is currently involved in autonomous systems and foveal vision, application of Artificial Neural Networks to Energy Management Systems, and in BroadBand and Multimedia Communication.