A Multi-layer Description of the Environment using Curvature Information for Robot Navigation

J. Almeida, L. J. Manso, A. Bandera* and P. Núñez

***Abstract*—In this paper, a novel method for a multi-layer description of the enviroment based on curvature information is presented. The proposal consists of three consecutive stages: data acquisition and pre-processing, segmentation and landmarks extraction. The main novelty of this work is the use of a RGBD sensor that projects the 3D points onto a set of planes to different heights. Each one of these planes corresponds to different scan laser readings that are segmented using an adaptive curvature estimation. This adaptive segmentation is used to directly extract breakpoints, line segments and real and virtual corners of the 3D environment. Experimental results show that the proposed approach is efficient for detecting landmarks in structured and semi-structured real and virtual environments. A comparative study of similar approaches in terms of robustness and accuracy demostrates the improvements of the presented environment description system.**

***Index Terms*—Adaptive curvature function, environment description, natural landmarks.**

I. INTRODUCTION

A mobile robot working in real scenarios must have the capability of autonomously navigating in its environment. One of the necessary prerequisites for autonomous navigation is the capability of building a map or a representation of the working area ($i.e.$, mapping). Extracting this spatial information from the environment has an important positive effect on different problems in robotics, such as localization, path planning, obstacles avoidence or Simmultaneous Localization and Mapping (SLAM).

In theory, if a robot is equipped with sensors, it can acquire an incredible amount of information about the spatial structure of its environment. Vision or range sensors ($e.g.$, sonar or laser) have been tradittionally used in the last decades for most of the researchers. Recently, low-cost RGBD sensors that acquire both RGB and depth images have become very popular in the robotics community. Independently of the type of sensor, the robot must choose the spatial representantion and be capable to process the reading at real time.

This paper adopts an approach based on features (or landmarks) for the spatial representation using a RGBD sensor, where landmarks can be defined as "distinct features that a vehicle can recognize reliably from its sensor observations". As general rule, the success of the environment description systems for robot navigation depends on: i) the chosen type of landmarks and the existence of accurate sensors capable of acquiring an enough number of features for a complete description of the environment; and ii) the availability of fast and reliable algorithms capable of extracting and characterizing landmarks from a large set of noisy and uncertain data.

The **main contribution** of this work is to provide the robot with the capacity to

J. Almeida, L. J. Manso and P. Núñez are members of Robotics and Artificial Vision Lab. *Robolab* Group, University of Extremadura, Spain.

*A. Bandera is with Electrical Technological Department, University of Málaga, Spain.
Author e-mail: pnuntru@unex.es

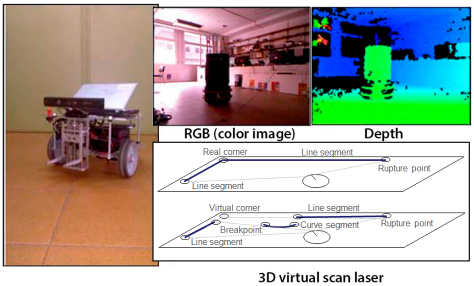

Fig. 1. The approach uses the RGBD sensor mounted on robot RobEx to extract and characterize a set of natural landmarks: line and curve segments, and real and virtual corners. The RGBD information is projected to different planes and each one is used like a virtual scan reading.

extract several types of landmarks that are present in structured and semistructured 3D indoor environments using a RGBD sensor at real-time. Information acquired by the sensor is projected onto a set of planes to different heights that are then segmented using an adaptive curvature estimation (see Fig. 1). This method extends the works presented by the authors in [1], improving the map generated by the robot and including more features in the 3D robot surrounding for being used in later robot navigation algorithms, such as SLAM or scan matching.

The presented paper is organized as follows: after discussing known approaches to the mapping problem for robotic navigation in Section II, Section III introduces the method for emulating different laser scans to different heights. Next, Section IV presents an overview of the proposed approach for describing the 3D robot environment. In Section V, the experimental results are pointed out, and finally, Section VI describes the conclusions and future works of this work.

II. Related Works

The problem of describing the robot surrounding has been tackled of different ways in the literature. Typical choices for the spatial representation include Topological [2], Grid-based [3], Semantic-based [4] and Feature-based [1], [5] approaches. Each representation has its own advantages and disadvantages. Typically Grid-based models are computationally expensive and require a huge amount of memory, whereas Topological-based and Semantic-based models rely on predefined scenarios, which assumes that some structures in the environments are known in advance ($e.g$, corridors or rooms). On the contrary, Feature-based representations allow the use of multiple geometric models to describe the different parts of the robot environment. In general, these methods reduce the computational load and memory constraints and makes them attractive because of their versatility and compactness.

Most of the Feature-based approaches are based on structures whose nature differs according to the environment ($e.g.$, indoor or outdoor) and the kind of sensor the robot is equipped with. Vision-based sensors have been used for extracting different features from the robot surrounding (a good review is done in [7]), however applying vision to the mapping problem leads to increase CPU usage due to the complexity of the algorithms. Range sensors, such as sonars or laser range finders, have been also used for extracting features from the environment [1], [5]. Contrary to vision systems, the complexity of the feature extraction algorithms that work with sonar or laser sensors is usually very reduced. Finally, RGBD sensors allows the robot to describe the environment using both image and depth information, processing colored clouds of 3D points whose computational load is very high [6]. The approach described in this paper works with depth information, but different to other methods, the robot uses the depth information from the RGBD sensor for generating a set of virtual laser range finder sensors whose 2D readings are after independently processed.

Structured or semi-structured indoor environments can be modeled using geometric features. Walls, columns or the furniture are able to be represented as a set of line segments, curve segments or corners. Given a range reading, many algorithms have been proposed for mobile robotic mapping using line features extracted from 2D range data. The work presented in [8] provides an interesting survey and comparison on line extraction algorithms for indoor environments, including the Split-and-Merge, Incremental, Hough-transform, Line regression, Random Sample Consensus (RANSAC), and ExpectationMaximization (EM) algorithms. Recently, in [9], the Distance-based Convolution Clustering (DCC) was introduced for grouping the robots scanned points into some clusters using a convolution operation. These clusters were used for detecting lines with a combination of Hough Transform and line tracking algorithm. In Mohamad et al.'s work [10], the authors define the Consecutive Clustering Algorithm (CCA) that incrementally adds new lines to the previously calculated map-lines and merges them with the overall map according to a statistical-based analysis. New approaches as the presented in [11], adopt a new methodology for detecting line and circle features that does not depend on prior knowledge of the environment.

All previous approaches process the 2D scan for finding a set of geometric features. Other approaches, such as the presented in [12] or most recently [1], analyze the laser scan as a local descriptor which can be processed to extract a set of dominant points which correctly segments the scan into curve and line segments. In these approaches, instead of using a slow, iterative solution, dominant points are robustly detected by adapting the scale to the local surroundings of each range reading. The work presented in this paper uses the adaptive curvature approach described in [1], that allows to robustly segment the virtual laser scans ($i.e.$, projections of the depth information onto different 2D planes) into curve and line segments and corners (real and virtual) in a fast way.

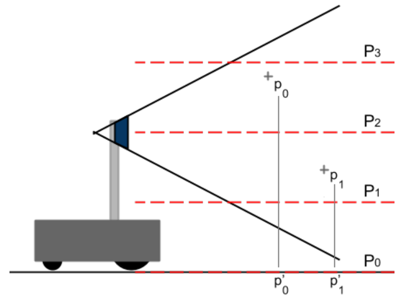

Fig. 2. Demonstration of how can the frustum of an RGBD sensor be divided given four input planes (P_0, P_1, P_2 and P_3) and how to use such planes to project two input points (p_0, p_1). The point p_0 would correspond to the pointcloud subset C_2, p_1 would correspond to C_1.

III. Virtual 3D laser from RGBD information

The generation of the virtual array of lasers is a three-stage process: a) split the point cloud gathered by the sensor in multiple point subsets separated by planes parallel to the ground and themselves; b) project all points of all subsets to the ground plane; and c) generate virtual laser readings with each of the subsets of points.

The separation and projection is straightforward: since the planes are parallel to the ground, the points can be partitioned using their height and the one of the different planes. Similarly, projection is achieved by setting the height of the points to zero. Once the points have been partitioned and projected –generating N smaller point clouds $C_0 ... C_N$, where C_i contains all the points between planes P_{i-1} and P_i (see Fig. 2)– the next step is to generate the virtual laser scans $L_0 ... L_N$: First, the virtual readings $p_i(j)$ are initialized with the maximum virtual laser distance.

$$\forall i, j : i \in [0, N), j \in [0, M) \; p_i(j) = \mathcal{M}$$

where M is the number of readings per laser scan and $\mathcal{M}$ the maximum distance (which depends on the RGBD sensor and its position). After initialization, for each simulated laser its output reading is updated using all

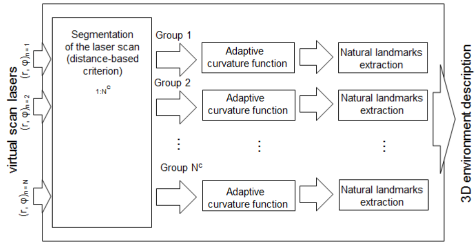

Fig. 3. Overview of the three stages environment description algorithm proposed in this paper.

the points in it. Each point p is converted to polar coordinates, so each of them can be matched to one specific reading of the virtual laser: if the distance of the point is smaller than the one of the current reading's value, the value is updated with the distance to the point.

Therefore, the information provided by a virtual laser sensor n in a single scan is typically in the form $\{(r,\varphi)_l | l = 1...N_R\}_n$, on which $(r,\varphi)_l$ are the polar coordinates of the l-th range reading (r_l is the measured distance of an obstacle to the sensor rotating axis at direction φ_l). Similar to real laser range finder, the virtual scan measurements are acquired by the virtual sensor with a given angular resolution $\Delta\varphi = \phi_l - \varphi_{l-1}$, which is previously chosen by the user. 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.

IV. Multi-layer curvature description of the environment

In this paper, the information acquired by a RGBD sensor is processed to emulate different laser scans on different heights. Each virtual laser reading is then analyzed according to the natural landmark extraction algorithm described in [1]. Therefore, the proposed approach provides a 3D description of the environment, where rupture points and breakpoints [13] and three features of interest: line segments, corners (real and virtual) and curve segments are detected over each virtual scan laser. An overview of the approach is illustrated in Fig. 3. A definition of each item is given below:

- Rupture points are measurements from the virtual scans associated to discontinuities due to the absence of obstacles in the scanning direction.
- Breakpoints are scan discontinuities due to change of surface being scanned by the emulated laser sensors.
- Line segments result from the scan of planar surfaces (e.g. walls or the furniture in indoor environments).
- Real Corners are due to change of surface being scanned or to change in the orientation of the scanned surface. Virtual Corners corrresponds to the intersection of two line segments.
- Curve segments result from the scan of curve surfaces (e.g. columns within indoor enviroments).

The method is summarized in the next subsections.

A. *Data acquisition and pre-processing*

Once a set of L_N virtual scan readings is provided by the RGBD sensor as was described in Sec. III, the first stage independently divides each laser scan into clusters of consecutive range readings according to a distance criterion. Let $(r,\varphi)_{i-1}$ and $(r,\varphi)_i$ be two consecutive range readings, they belong to the same segment iff the distance between them is less than a given threshold. Despite using a fixed threshold, these segment boundaries (breakpoints) are extracted using the adaptive breakpoint detector [13]. In Fig. 4b, the segments associated to different clusters for two different scan lasers in the scenario drawn in Fig. 4a are shown with different colors.

B. *Segmentation of the virtual scan laser based on curvature information*

In the next stage, the curvature information is used to characterise the local planar scan provided by each virtual laser sensor.

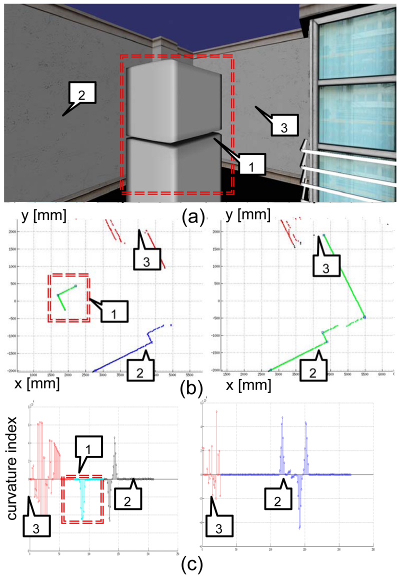

Fig. 4. a) Simulated environment; b) Segments of two virtual laser scans (different colors are associated to distinct clusters); and c) curvature functions associated to b) (see Section C).

The curvature index at each range reading of the laser scan is adaptively filtered according to the distance between possible corners in the whole laser scan. This method permits to remove noise, but scan features are nevertheless detected despite their natural scale. For each range reading i of a laser scan, the Cartesian coordinates are evaluated. The proposed method for adaptive curvature estimation in laser scan data consists of the following steps:

1) Calculation of $K_f(i)$ and $K_b(i)$, the maximum length of laser scan presents no discontinuities on the right and left sides of the range reading i respectively. $K_f(i)$ is calculated by comparing the Euclidean distance from i to its $K_f(i)$-th neighbour ($d(i, i + K_f(i))$) to the real length of the laser scan between both range readings ($l(i, i + K_f(i))$). These distances tend to be the same in absence of corners, even if laser scans are noisy. Otherwise, the Euclidean distance is quite shorter than the real length. Thus, $K_f(i)$ is the largest value that satisfies

$$d(i, i + K_f(i)) > l(i, i + K_f(i)) - U_k \quad (1)$$

where U_k is a constant value that depends on the noise level tolerated by the detector. $K_b(i)$ is also set according to Eq. (1), but using $i - K_b(i)$ instead of $i + K_f(i)$. In the presented work, it has been experimentally proven that U_k equal to 1.0 works correctly.

2) 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 $i + K_f(i)$ and between i and $i - K_b(i)$. If (x_i, y_i) are the coordinates of the range reading i, the local vectors associated to i are defined as

$$\begin{aligned} \vec{f_i} &= (x_{i+K_f(i)} - x_i, y_{i+K_f(i)} - y_i) \\ &= (f_{x_i}, f_{y_i}) \\ \vec{b_i} &= (x_{i-K_b(i)} - x_i, y_{i-K_b(i)} - y_i) \\ &= (b_{x_i}, b_{y_i}) \end{aligned} \quad (2)$$

3) Calculation of the curvature index $|K_\theta(i)|$ (*i.e.*, angle associated to the range reading i). The angle at range reading i can be estimated as follows

$$|K_\theta(i)| = arccos\left(\frac{\vec{f_i} \cdot \vec{b_i}}{|\vec{f_i}| \cdot |\vec{b_i}|}\right) \quad (3)$$

4) Detection of corners over $|K_\theta(i)|$. The obtained curvature index represents the curvature associated to each range reading in an absolute manner. Corners are those range readings which satisfy the following conditions: i) they are local peaks of the curvature function and ii) their $|K_\theta(i)|$ values are over the minimum angle required to be considered a corner instead of a spurious peak due to remaining noise (θ_{min}).

Fig. 4c illustrates the curvature functions associated to the virtual scan lasers shown in Fig.4b.

C. Natural landmarks extraction from the curvature function

The adaptive curvature function $|K_\theta(i)|$ can directly provide three different natural landmarks. In order to include these items as landmarks in later robotic tasks, it is necessary to characterize them. Next, $|K_\theta(i)|$ is analyzed to extract the set of features of the robot surrounding, that is achieved by fitting parametric curves to the measurement data associated with each line or curve segment.

1) Detection of line segments over $|\kappa_\theta(i)|$. Line segments result from the scan of planar surfaces. Therefore, they are those sets of consecutive range readings which: i) are under a minimum angle (in the proposed experiments, this minimum curvature height, θ_{min}, has been fixed at 0.05); and ii) have a size over a minimum length value (l_{min}=10 range readings). In order to characterize the line segment, the method described in [1] is used. This method uses a linear regression that approximate the set of points to a straight-line in slope-intercept form (α, the angle between the x-axis and the normal of the line, and d the perpendicular distance of the line to the origin).
2) Detection of curve segments over $\kappa_\theta(i)$. Curve segments result from the scan of curve surfaces. Contrary to the curvature values associated to a line segment, it can be appreciated that the curvature function associated to a curve segment presents a consecutive set of local peaks, some of them could be wrongly considered as corners. To avoid this error, the algorithm associates a cornerity index to each set of consecutive range readings whose $\kappa_\theta(i)$ values are over θ_{min} or under $-\theta_{min}$ and have a size over l_{min}. This cornerity index, ci, is defined as

$$ci = \frac{\frac{1}{i_e - i_b}\sum_{j=i_b}^{i_e} \kappa_\theta[j]}{max_{i\in(i_b,i_e)}\{\kappa_\theta[i]\}} \qquad (4)$$

where i_b and i_e are the range readings that bound the possible curve segment. If ci is close to one, the mean curvature of the segment and its maximum value are similar, and the segment can be considered as a curve segment. If ci is low, the mean curvature of the segment is lower than its maximum value. Then, the segment cannot be considered as a curve segment. Therefore, 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 (U_c has been fixed at 0.5 in all experiments). Finally, the cluster of points is then fit to a circle ($i.e.$, center in (x_0, y_0) and radius ρ).
3) Detection of corners over $\kappa_\theta[i]$. Corners are easily detected from the analysis of the curvature function as a value associated to a local peak of the curvature function, and a region bounded by two range readings, i_b and i_e. Thus, $\kappa_\theta[i]$ values must be over the minimum angle required to be considered a corner instead of a spurious peak due to remaining noise. Once a real corner is detected, its position (x_c, y_c) is estimated as the intersection of the two lines which generate it. Virtual corners are defined as the intersection of extended line segments which are not previously defined as real corners. Therefore, virtual corners are also characterized in the same way that real corners.

Fig. 5 show the natutal landmarks extracted by the proposal from the virtual scan lasers presented in Fig. 4b.

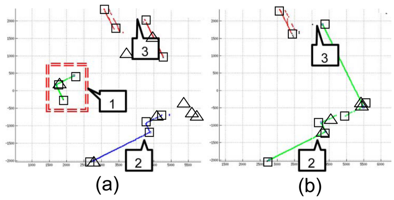

Fig. 5. a-b) Landmark detection associated with Fig. 4 (square - line segments end-points, triangle - corners).

V. Experimental results

To evaluate the performance of the proposed algorithm, real and simulated data were used. The algorithms have been developed in C++ software and the benchmark tests have been performed on a PC with processor Intel Core i5 2.4GHz with 4Gb of DDR3 RAM and GNU-Linux Ubuntu 13.10. The experiments were focused on the evaluation of the algorithm in terms of robustness, number of detected features and computational resources. A comparative study of the proposal with the previous environment description algorithm presented in [1] is also provided.

A. *Software architecture*

The software to control the proposed system is built on top of the robotics framework RoboComp [14]. The software architecture consists of a new component laserRGBDComp implemented in the framework, which fuses the RGBD information (rgbdComp) into different emulated scan lasers running in parallel. The software component cubafeaturesComp has been modified in order to add 3D scan laser readings.

B. *Simulated environments*

The simulated data were acquired using RCIS (RoboComp InnerModel Simulator) [14]. RCIS is a 3D robot simulator designed for use in academia, in early stages of development, and, mainly, for research purposes. One of its most remarkable features is that it enables users to control the noise produced by the simulated sensors and actuators. This ability can be used to test how robust algorithms are against noise and, if noise is set to zero, to differentiate between problems dealing with noise and algorithm errors. RCIS implements all the interfaces of the hardware abstraction layer of RoboComp (e.g., Camera, Laser, RGBD, Joint Motor Bus), which represent most of the common hardware used in autonomous robotics. For the tests described in this paper different RGBD sensors with different noise levels were simulated within an indoor scenario. The use of this simulator allows an evaluation of the algorithms according to different sensor noise values. Fig. 6a illustrates the simulated environment used in the experiments. The scenario consists of a simple room and a cylinder (marked as 1 in the figure) and two boxes of different heights (2 and 3 in Fig. 6a) located in it. In Fig. 6b, the segmentation of the robot surrounding using five different virtual laser sensors is shown. Finally, natural landmarks detected by the algorithm for the virtual scan lasers L_1 and L_3 are drawn in Fig. 6c. Line segment end-points, corners and circle segments are illustrated as squares, triangles and circumferences, respectively.

The simulated scenario has been used for evaluating the number of landmarks detected by the proposed algorithm and its dependence with the error of the RGBD sensor and the number of virtual laser scans. These results are summarized in Table I and Table II. As shown in Table I, the computational load allows to work at real time (all the components run in parallel) and the number of features is considerably increased (n_c, n_{ls} n_{cs} and n_t are the numbers of corners, line segments, curve segments and total features detected by the algorithm, respectively). In [1], the robustness of the environment description algorithm was evaluated according to two different metrics ($TruePos$ and $FalsePos$) and compared to similar approaches with better results. Therefore, for this comparative study, the presented

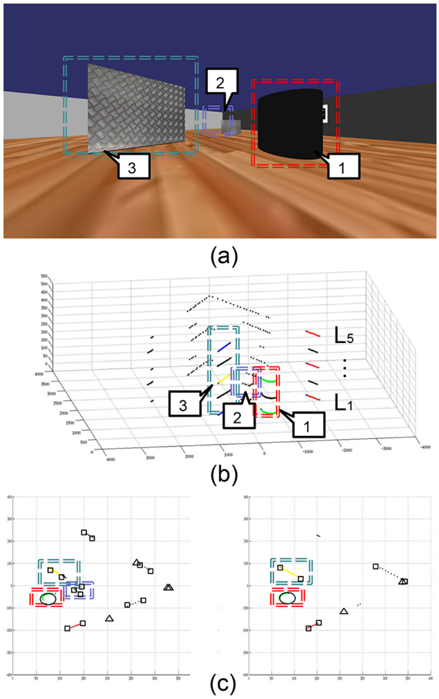

Fig. 6. a) Simulated scenario used in the experiment; b) virtual scan lasers (L_i $|i = 1...5$); and c) natural landmarks extracted by the proposal for two different scan laser, L_1 and L_3.

approach is only compared to the previous work. Thus, one virtual scan has been taking into account in a fixed height (L_3) and different noise levels are added to the RGBD information. Three different RGBD sensor noise values were used in the experiments. The distance noise was simulated using a normal distribution with μ = 0, and variance values 0, 0.01 and 0.02 meters, respectively. Table II updates the results described in [1], including the results of the presented proposal. Obviously, the robustness of the proposal is similar to the other one and the differences are insignificant and due to the cumulative effect of using more planes.

C. Real environments

Real data were acquired using a RobEx robot (Figure 1). The robot RobEx is a differential robot designed by RoboLab at the University of Extremadura. For the experiments described in this paper, a Kinect was mounted on top of it and set up for acquiring RGBD images at 30 fps. The robot was teleoperated to a test area whitin RoboLab facilities. Boxes and false walls were located in front of the robot. Fig. 7a shows the test area. In Fig. 7b, five different virtual scan lasers are drawn. An example of the extracted features from the L_3 virtual scan laser is shown. The number of landmarks has been considerably increased respect to only using a real scan. Table III shows the results of the algorithm in this scenario.

TABLE I
EXPERIMENTAL RESULTS OF THE ALGORITHM.

time consumption (ms)	n_c	n_{ls}	n_{cs}	n_t	L_i
6.0	5	7	1	13	L_1
6.0	5	7	1	13	L_2
5.9	2	3	1	6	L_3
5.9	2	3	0	5	L_4
5.9	1	3	0	4	L_5
5.95	**15**	**23**	**3**	**41**	$\mathbf{L_{N=5}}$

TABLE II
COMPARATIVE STUDY WITH THE PREVIOUS AUTHOR'S WORK (SEE [1])

Núñez et al. ([1])	Proposed ($L_{N=5}$)	Algorithm
0.875	0.825	TruePos (σ=0m)
0.0	0.03	FalsePos (σ=0m)
0.815	0.79	TruePos (σ=0.01m)
0.02	0.12	FalsePos (σ=0.01m)
0.750	0.70	TruePos (σ=0.02m)
0.8	0.18	FalsePos (σ=0.02m)

TABLE III
RESULTS OF THE ALGORITHM IN THE REAL ENVIRONMENT ($L_{N=5}$)

n_c	n_{ls}	n_{cs}	n_t
27	20	0	**47**

VI. CONCLUSION AND FUTURE WORKS

This paper presents a multi-layer description of the robot surrounding by using a set of virtual scan lasers to different heights from a low-cost RGBD sensor. These virtual scan readings are processed for extracting natural landmark from the environment in basis to its curvature information at real-time. This approach extends the previous

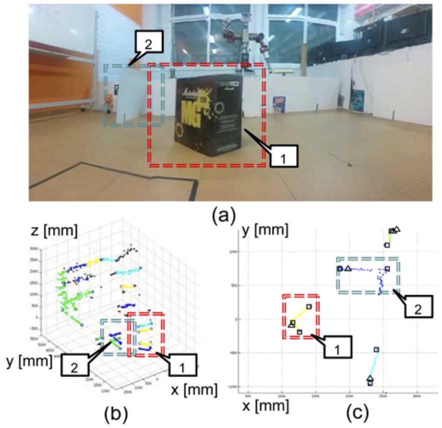

Fig. 7. a) Real scenario used in the experiment; and b) natural landmarks extracted by the proposal using five different scan lasers.

work of the authors to 3D scenarios, increasing the number of landmarks detected and characterized by the algorithm. The method has been evaluated in real and simulated scenarios and compared to previous approaches. Futute works are focused on using these natural landmarks in more complex robotics task, as localization or SLAM. Also the color information from the RGBD sensor can be evaluated in the segmentation process.

Acknowledgment

This work has been supported by the MICINN Projects TIN2012-38079-C03-01 and TIN2012-38079-C03-03 and by the Extremadura Project GR10144.

References

[1] P. Núñez, R. Vázquez-Martín, J.C. del Toro, A. Bandera and F. Sandoval, "Natural landmark extraction for mobile robot navigation based on an adaptive curvature estimation". Robotics and Autonomous Systems, Vol 56, (3), pp. 24–264, 2008.

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

[3] B. J. Kuipers, The spatial semantic hierarchy, Artificial Intelligence 119, pp. 191-233, 2000.

[4] A. Nuchter and J. Hertzberg, Towards semantic maps for mobile robots, in Journal of Robotics and Autonomous Systems (JRAS), Special Issue on Semantic Knowledge in Robotics, vol. 56, no. 11, pp. 915 926, 2008.

[5] J. D. Tardós, J. Neira, P. M. Newman and J. J. Leonard, "Robust mapping and localization in indoor environments using sonar data", Int. Journal of Robotics Research, pp. 311-330, 2002.

[6] P. Henry, M. Krainin, E. Herbst, X. Ren and D. Fox. "RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments" in International Symposium on Experimental Robotics, 2010.

[7] N. Muhammad, D. Fofi, and S. Ainouz, "Current state of the art of vision based SLAM" in Proceedings of the SPIE, article id. 72510F, 12 pp, 2009.

[8] V. Nguyen, S. Gachter, A. Martinelli, N. Tomatis, R. Siegwart, "A comparison of line extraction algorithms using 2D range data for indoor mobile robotics", in Autonomous Robots 23 (2) (2007) 97111

[9] C. Fernndez, V. Moreno, B. Curto, J.A. Vicente, "Clustering and line detection in laser range measurements", in Robotics and Autonomous Systems 58, pp. 720-726, 2010.

[10] M. Movafaghpour and E. Masehian, "Poly line map extraction in sensor-based mobile robot navigation using a consecutive clustering algorithm", in Robotics and Autonomous Systems, vol. 60, pp. 107–1092, 2012.

[11] Y. Zhao and X. Chen, "Prediction-based geometric feature extraction for 2D laser scanner", in Robotics and Autonomous Systems, Vol. 59, pp. 40–409, 2011.

[12] R. Madhavan, H.F. Durrant-Whyte, "Natural landmark-based autonomous vehicle navigation", Robotics and Autonomous Systems, Vol. 46, pp. 79-95, 2004.

[13] G.A. Borges, M. Aldon, Line extraction in 2D range images for mobile robotics, Journal of Intelligent and Robotic Systems Vol. 40, pp. 267-297. 2004.

[14] L.J. Manso, P. Bachiller, P. Bustos, P. Nuez, R. Cintas and L. Calderita, RoboComp: a Tool-based Robotics Framework. In Simulation, Modeling and Programming for Autonomous Robots (SIMPAR), pp. 251–262. 2010.