

A Cost-Efficient 3D Sensing System for Autonomous Mobile Robots

Marco A. Gutiérrez, E. Martinena, A. Sánchez, Rosario G. Rodríguez, P. Núñez.

Abstract— This paper describes a mechanism for building an inexpensive and, at the same time, accurate system for 3D scanning on Autonomous Mobile Robots. Our system allows us to obtain 3D points from the robot environment along with its associated color. This data can be later processed using different techniques in order to obtain information from surrounding objects useful for tasks such as navigation or localization. Information is obtained at a rate of 50 ms per line of scan (700 points per line). In order to use the sensor as part of an active perception system, resolution is made to be directly dependent on the scanning speed and robots are able to adjust the related parameters accordingly to their needs. Our approach uses a regular commercial 2D Laser Range Finder (LRF), a step motor and a camera, all this controlled by an embedded circuit which makes the system apt for being built in any regular Autonomous Mobile Robot. Finally, to test our system, two different real applications have been used. First a 3D Map reconstruction is done using several point clouds matched by the ICP algorithm and our odometry. Then, we make a novelty detection and 3D shape retrieval using the Gaussian Mixture Model and Superquadrics.

Index Terms—Autonomous Mobile Robots, 3D Shape retrieval, Mapping, Laser Range Finder, RGB-D

I. INTRODUCTION

Sensing and processing the unknown environment is crucial for Autonomous Mobile Robots to obtain information from their surroundings. Most of the actions a mobile robot could achieve, such as mapping, localization, exploration or navigation, have strong dependences on the information obtained from their environment. Hence, the task of properly acquiring and processing this information has become a critical need in the mobile robotics field.

In order to obtain this information Autonomous Mobile Robots, can use different sensing systems to achieve data. Cameras are one of the most common used interfaces to obtain environment data. They have been widely studied and, therefore several algorithms are available to achieve world modeling. For this reason, not only direct information is now obtained from cameras, other, such as depth, can be estimated following several methods [1], [2], [3], [4], however most of these solutions have a necessity for texture information. Therefore, these algorithms have a strong dependency on light variations and wouldn't work properly in indoors environments specially those with surfaces not uniformly colored. LRF (Laser Range Finder) solutions provide a more accurate choice in terms of precision and environmental dependency. However, they lack from texture information when it is available.

Marco A. Gutiérrez, E. Martinena, A. Sánchez, Rosario G. Rodríguez and P. Núñez are with University of Extremadura.
E-mail: marcog@unex.es

The lack of good, cheap and fast sensors allowing robots to sense the environment in real-time, in order to allow them to act on the basis of acquired data, is one of the reasons of the gap between prognoses and reality. Several commercial accurate 3D LRF systems are already available in the market. However, most of them have usually a high cost (~ 50.000 USD). This way, we have developed a 3D sensing system for Autonomous Mobile Robots. It consists on a 2D LRF moved by a step motor, a camera for texture information and an embedded system that manages the other three components. Although this managing choice could limit, in some way, the whole system it has been chosen that way because it is intended to be used in Autonomous Mobile Robots and they have a strong needs for small and power-efficient sensing systems. The embedded system is in charge of directly moving the step motor, acquiring information from the LRF and the camera, collecting the 3D data and sending it over the network for its processing, storage or visualization.

This paper is organized as follows. In section II we talk about some previous works in the field. Our 3D sensing system is deeply explained in Section III, its hardware components and how the data is processed. Then we test our system using two different applications in Section IV. First building a 3D map of the robot environment using the Iterative Closest Point (ICP) algorithm and the odometry information, and then detecting changes on 3D maps applying the Gaussian Mixture Model and retrieving shapes from detected objects using superquadrics. Finally, in Section V we present some conclusions and future work directions.

II. PREVIOUS WORKS

Autonomous Mobile robots have a strong need of 3D ranging sensors. Therefore researchers have been increasingly focusing efforts in this field, and, as a consequence, groups have come up with different approaches to obtain 3D points and shapes from the environment.

The relative low cost of cameras have made these systems very common. Therefore, using cameras to obtain 3D information has been a widely spread research effort. In order to obtain depth information from images the number of cameras used have changed along the different studies. The most popular solution is using two cameras (Binocular Stereo), mostly inspired in the way humans obtain depth perception from their environment. Several groups have done intensive works on these systems [2] [3] [4]. But also other number of cameras such as three [1], one (monocular cameras) [9] or even only using unsorted collections of images [10], have been

used. However, these systems are highly dependent on texture information. This makes them to, usually, loose accuracy when facing indoor environments. In the same way, most of the time, these solutions have high computational costs with big power requirements and a small field of view (usually 60°).

In order to solve the lack of texture dependency the use of laser lines projection have taken into account [8]. Even some RGB-D commercial sensors widely popular nowadays such as the Primesense RGB-D sensor [20] make use of infrared projection to reduce this texture need. Although solutions are very popular, compared to LRF performance, they get small fields of view, low depth precision (3cm in 3m scan) and high sensibility to light variations.

Due to the price difference from commercial 2D LRF to 3D LRF, solutions to achieve the whole 3D spectrum with 2D LRF systems have been explored. This has been done ether making the 2D LRF scanner rotate itself [12] or rotating a mirror in front of the it [13]. Some of these solutions even include a camera for texture information retrieval [14]. On our system, efforts have been not only focused on the LRF-camera 3D system but also on managing it from an embedded system in order to make it able for running on small and low power consumption Autonomous Mobile Robots.

Finally, some other different 3D retrieval solutions are worth to mention, like using a rotating camera and an angled LRF [15] or even using camera shadows to construct Multiple Virtual planes [11].

III. SYSTEM DESIGN

To develop a complete sensing system that can cover most of the mobile robotic possible needs, a wide range of possibilities must be taken into account. Indoor or outdoor scenes, more or less light or uniformly or not uniformly colored objects are some of the different aspects from the environment a robot can face. Our device is made up in an effort to take into account all these likelihoods, trying to get the best and more accurate info out of each situational environment.

Our design is intended to be simple and low cost ($\sim 3500\text{€}$). And although, some of the hardware could be cheaper (laser is $\sim 3000\text{€}$), it has been chosen that way due to our target of getting the most available information out of the surroundings a robot could find itself, while, at the same time keeping it small and power-efficient to be deployed on Autonomous Mobile Robots.

A. Hardware

Our solution consists of three main hardware parts, showed in Fig. 1. First, a commercial 2D LRF is moved by a step motor (labeled as (1) and (2) on Fig. 1) to be able to obtain 3D points from full 360° scans. Secondly, a camera (labeled (3) on Fig. 1) is used in order to achieve texture feedback from the environment. When this information is made available, we are able to attach color information to the correspondent points obtained from our LRF. And finally, an embedded system (labeled (4) on Fig. 1) that manages the rest of the components in the system. Fig. 1b shows a sketch of the system and the associated reference frames of some of the elements.

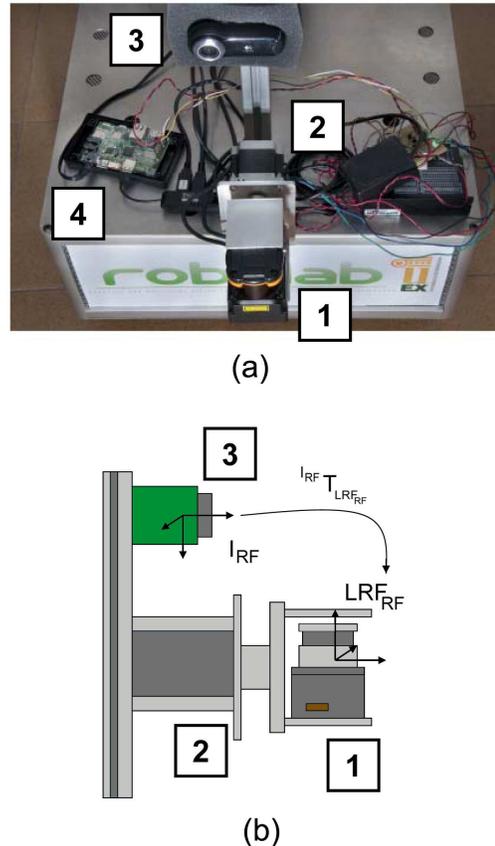


Fig. 1: a) The 3D sensing system mounted on one of our Autonomous Mobile Robots, *Robex*. The three main hardware parts are marked in the Figure (LRF (1), step motor (2), RGB camera (3) and embedded system (4)) b) Sketch of the measurement system and associated reference frames.

We use a Hokuyo LRF with a large scanning range (30 meters and 270°). It has been chosen that way so that we can make scans ether indoor or outdoors with appropriated accuracy, although the system is fully compatible with almost any other 2D LRF sensor. The LRF is attached to a step motor to make it scan the full 360° in front of the mobile robot. The step motor has enough torque to move the LRF without using any gear-train and, in consequence avoiding the backlash they usually introduce. It has 200 steps resolution and it is attached to a power driver to obtain a higher one, up to 25000 steps. Between the several ways of moving a laser that exist, we have chosen to do it in the Z axis, with $\alpha=0$ degrees, as shown in Fig. 2. This is the solution that better fits our needs because it leads to high resolution of points in front of the robot, exactly the place it is facing and most possibly to where it is moving [7]. This allows us to focus our density distribution on certain objects, i.e. obstacles or new elements introduced in the environment [18]. The resolution we want to achieve on the scan is directly dependent on the speed of the step motor. Therefore, our system is good to be used for active perception purposes since most of the resolution is kept

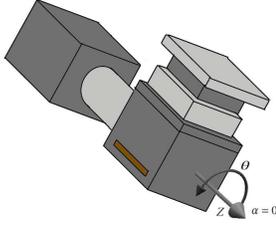


Fig. 2: LRF and step motor rotating scheme.

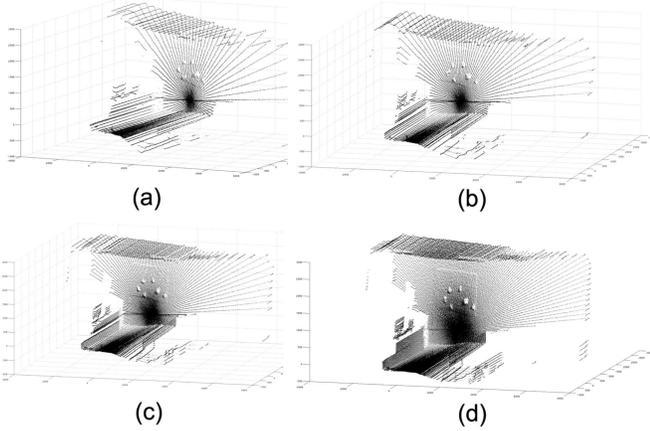


Fig. 3: Scans using different speeds, and therefore achieving different levels of resolution: a) 0.6 rad/s, ~ 25 lines of scan b) 0.4 rad/s, ~ 50 lines of scan c) 0.2 rad/s, ~ 100 lines of scan d) 0.1 rad/s, ~ 200 lines of scan.

in front of the robot and is able to select the proper speed for each scan, changing the resolution in consequence. Fig. 3 shows different resolutions scans from our system, according to selected speed and its approximated scan resolution.

The Camera consists of a regular Web-Cam that provides us with 640x480 images at a rate of 10 frames per second. It is kept statically on top of the LRF and the step motor facing the same space as the scanner in order to match the color pixels from the camera with their correspondent 3D points from the LRF. The camera is USB connected to the embedded system to make more accurate and real-time matching of the captions and the 3D points. Both two sensors have different reference systems, I_{RF} and LRF_{RF} (see sketch at Fig. 1b). Therefore, as mathematically shown in Section III-B, data information from the camera would be calibrated, that is, processed to match the LRF one. This process involves finding the transformation ${}^{I_{RF}}T_{LRF_{RF}}$ (Fig. 1b).

Finally, all these elements are controlled by our embedded system. It has a *GNU/Linux* distribution and several *Robocomp* [5] components on top of it. It takes responsibility for properly moving the step motor according to given instructions and, therefore, assuring the required resolution. It makes calculations to retrieve the angles and data coming from the LRF. Besides, it captures images from the camera and assigns pixels to the corresponding 3D points. Then, information can be

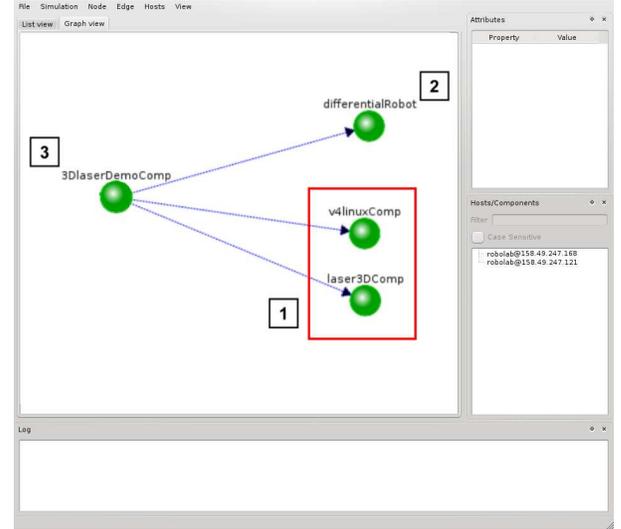


Fig. 4: Screenshot of the *managerComp Robocomp* tool. 1) The 3D laser and the Camera running on the embedded system 2) The component in charge of the odometry running on the robot 3) The process and display component running on a Desktop system.

locally stored for later processing, or sent through the network for live treatment, storage and/or display.

B. Data processing

The software to control our system is built in on top of the robotics framework *Robocomp* [5]. Making use of its component oriented programming and its communication middleware we are able to minimize the CPU load in our embedded system leaving the heavy processing for the powerful desktop systems. A light-weight camera component handler and another for the LRF and the step motor are executed directly in the embedded system (see (1) on Fig. 4), sending the data over the network to other components running on desktop computers that will make further data storage, analysis and/or display (see (3) on Fig. 4). Thanks to the communication middleware the system constitutes a generic component that can be used through its interface:

```

1 interface Laser3D
2 {
3   void moveToAndSweepAtSpeed (float
      minAng, float maxAng, float speed,
      bool once, int numScans, out int
      nScans, out float period);
4   void stop();
5   TLaserData getLaserData();
6   Laser3DConfData getLaserConfData();
7 }

```

Listing 1: Interface of the *Laser3D Robocomp* Component

1) *Coordinate systems conversion*: The 2D LRF returns points in the Polar Coordinate System (Fig. 5a) that is, a distance r and a polar angle or azimuth $\varphi \{(r_i, \varphi_i) | i = 1 \dots n\}$. Then, making use of the information coming from our step

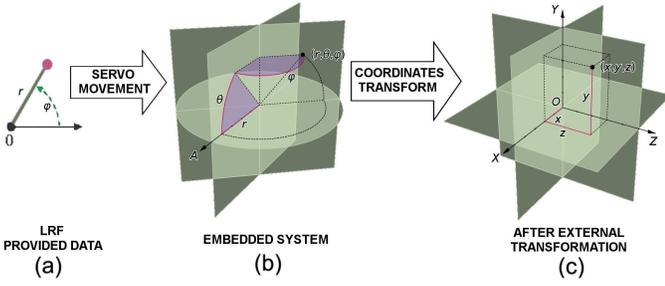


Fig. 5: Transformation from the LRF data to Cartesian Coordinates.

motor we obtain the inclination angle θ of the LRF. This leaves our coordinates expressed in the Spherical System (Fig 5b), having a radial distance r , an azimuth angle ϕ and an inclination angle θ , $\{(r_i, \phi_i, \theta_i) | i = 1 \dots n\}$. Then, data is finally converted into the Cartesian System $\{(x_i, y_i, z_i) | i = 1 \dots n\}$ (Fig. 5c), using the following regular systems conversion equation:

$$\begin{cases} x_i = r_i \cdot \sin\theta_i \cos\phi_i \\ y_i = r_i \cdot \sin\theta_i \sin\phi_i \\ z_i = r_i \cdot \cos\theta_i \end{cases}, \quad i = 1 \dots n \quad (1)$$

It is desired to keep the embedded system as much free of CPU load as possible to assure a good real-time response. However, since our inclination angle θ is directly dependent on the step motor movements information and we want to preserve the system accuracy, it is unavoidable but to execute the polar system coordinates to the spherical system transformation on our system. Therefore, transforms up to the spherical coordinates are processed on the embedded system, then data is sent over the network and the rest of the processing steps are computed on external and more powerful systems.

2) *Extrinsic 3D-LRF Camera Calibration*: Camera image data is also obtained. It comes in form of RGB matrix that is matched to its correspondent laser points. For this matching Rotation (R) and Transformation (T) of those color points to the laser reference frame must be performed [6]. In order to calibrate our camera-LRF, calculation of proper R and T is needed. This R and T are needed to obtain the target (X^L, Y^L, Z^L) , therefore we can say:

$$\begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix} = R \begin{bmatrix} X^L \\ Y^L \\ Z^L \end{bmatrix} + T \quad (2)$$

Where (X^c, Y^c, Z^c) are points on the reference system of the camera and (X^L, Y^L, Z^L) the ones from the laser. Also written as:

$$\begin{aligned} X^L &= r_{11}X^c + r_{12}Y^c + r_{13}Z^c + T_x \\ Y^L &= r_{21}X^c + r_{22}Y^c + r_{23}Z^c + T_y \\ Z^L &= r_{31}X^c + r_{32}Y^c + r_{33}Z^c + T_z \end{aligned} \quad (3)$$

We obtain n empirically known matching pair points in our LRF and camera, in the form:

$$((X_i^L, Y_i^L, Z_i^L), (x_i, y_i)), \quad i = 1 \dots n \quad (4)$$

Where every point (x_i, y_i) obtained from the image matches the corresponding (X_i^L, Y_i^L, Z_i^L) obtained from the LRF.

Following [6] we could conclude that using equation 5 we can obtain a solution.

$$Av = 0 \quad (5)$$

Where A is defined in equation 6 with an $n \geq 7$, and being the points not coplanar.

$$A = \begin{pmatrix} x_1 X_1^L & x_1 Y_1^L & x_1 Z_1^L & x_1 & -y_1 X_1^L & -y_1 Y_1^L & -y_1 Z_1^L & -y_1 \\ \vdots & \vdots \\ \vdots & \vdots \\ x_n X_n^L & x_n Y_n^L & x_n Z_n^L & x_n & -y_n X_n^L & -y_n Y_n^L & -y_n Z_n^L & -y_n \end{pmatrix} \quad (6)$$

This solution depends on a parameter or scale factor:

$$V = (v_1, \dots, v_8) \quad \begin{cases} v_1 = r_{21} & v_5 = \lambda r_{11} \\ v_2 = r_{22} & v_6 = \lambda r_{12} \\ v_3 = r_{23} & v_7 = \lambda r_{13} \\ v_4 = T_x & v_8 = \lambda T_x \end{cases} \quad (7)$$

Imposing the rotation matrix orthogonality we could determine the mentioned matrix and two components of the translation vector T_x and T_y . Then, to obtain the last component of the translation vector, T_z , we only have to solve equation 8.

$$B \begin{pmatrix} T_z \\ f_x \end{pmatrix} = b \quad (8)$$

Where:

$$B = \begin{pmatrix} x_1 & (r_{11}X_1^L + r_{12}Y_1^L + r_{13}Z_1^L + T_x) \\ \vdots & \vdots \\ \vdots & \vdots \\ x_n & (r_{11}X_n^L + r_{12}Y_n^L + r_{13}Z_n^L + T_x) \end{pmatrix} \quad (9)$$

and

$$b = \begin{pmatrix} -x_1(r_{31}X_1^L + r_{32}Y_1^L + r_{33}Z_1^L) \\ \vdots \\ \vdots \\ -x_n(r_{31}X_n^L + r_{32}Y_n^L + r_{33}Z_n^L) \end{pmatrix} \quad (10)$$

Then we obtain T_z following:

$$\begin{pmatrix} \hat{T}_z \\ \hat{f}_x \end{pmatrix} = (B^t B)^{-1} B^t b \quad (11)$$

This way we have obtained all the components of the correspondent Rotation Matrix R and Translation Vector T which are used to transform the points from one system to the other or, in the same way, assign the texture information to the 3D laser points (see Fig. 1b).

Fig. 6 shows a scan of our lab taken at $0.2\pi rad/sec$ of speed (10 seconds per 360° scan). The image is shown at Fig. 6a and data points from the same scan at Fig. 6b. After, as

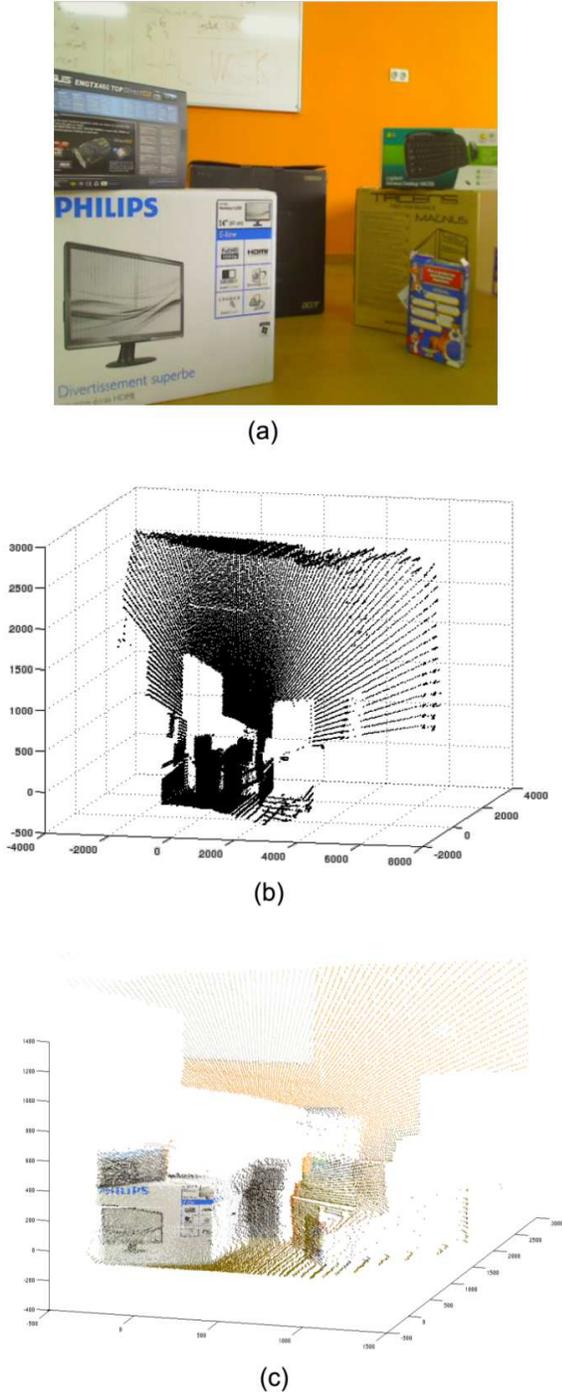


Fig. 6: (a) Camera image from a scan in our lab (b) Points from our 3D LRF (c) Colored data form the scan after data transformation.

explained, properly matching both data sets a result such as the one shown in Fig. 6c is obtained. Notice that colored data is not the whole 3D data from LRF, since the camera covers a much smaller area than the LRF, as explained on Section II.

IV. EXPERIMENTAL RESULTS

We have built our testing system using a Hokuyo UTM-30LX 2D LRF. It has 30 meters and 270° of scanning range. The LRF scans 270° at a rate of 25ms. However, we get the scans at a speed of 50ms due to the use of our, in some way, CPU limited embedded system. The 2D LRF is attached to a CeNeCe 23HB56 step motor to achieve the whole 3D spectrum. The motor has 200 steps and a 11 Kg/cm torque and it is attached to a Leadshine DRP452 driver with 15 different modes of resolution (from 1/2 to 1/125 steps). A Logitech Quickcam Pro 9000 Web Camera is used to obtain texture information from the environment. It gives us 640×480 images at a rate of 10 FPS (Frames Per Second). This all is controlled by our embedded system consisting of an ARM Cortex-A8 OMAP 3530 of 720 Mhz and 4G Mobile Low Power DDR SDRAM @ 200 Mhz running an Ubuntu GNU/Linux System. All of it is properly mounted and assembled into *Robex*, one of our Autonomous Mobile Robots (see Fig. 1).

The software have been developed using the *Robocomp* framework [5], which makes it easy to balance the work load between the embedded system and other more powerful ones. On the embedded system we have deployed two main components, one for managing the step motor and laser system and other for managing the camera. This data is served through the network to another external components running in more powerful desktop machines. Then, as said, the collecting data component ether processes, stores or shows data for its further analysis.

We have tested our 3D sensing system using two different algorithms. First we have run a scan matching in an effort to make a map of the environment of the robot. For this algorithm we have tried obtaining different scans at different points of the room and then matching them using ether the ICP scan matching algorithm [16] or odometry.

The second experiment consists of a novelty detection on a 3D map and a subsequent shape retrieval of detected novelty using superquadrics. We have used an algorithm that simplifies the data using a multi-scale sampling technique in order to reduce the computation time of detecting changes in the environment. Then a method based on the Earth Mover's Distance (EMD) and Gaussian Mixture Models (GMM) [18] is used to detect changes and obtain a segmented point cloud representing those changes. Finally the 3D shape of the object is retrieved using a superquadric approximation to the point cloud.

A. Mapping

Several tests in the mapping field have been performed. In one of them, we have used the Chen-Medioni (point-to-plane) framework for ICP (Iterative Closest Point) [16]. Having a collection of points (p_i, q_i) with normals n_i this algorithm tries to determine the optimal rotation and translation to be applied to the first collection of points, i.e. p_i to bring them into alignment with the second q_i . It obtains a rotation R and translation t trying to minimize the alignment error:

$$\varepsilon = \sum_i [(Rp_i + t - q_i) \cdot n_i]^2 \quad (12)$$

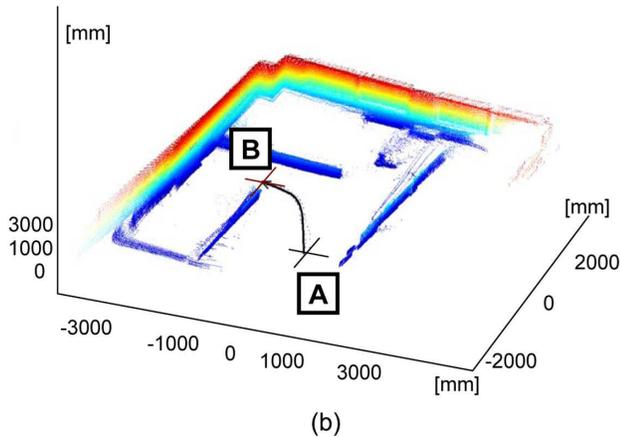
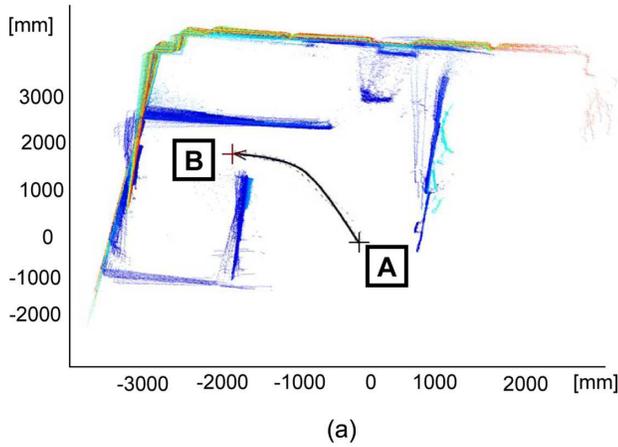


Fig. 7: ICP Scan Matching using 24 different scans and moving from A to B an angle of $\beta \leq 0.2 \text{ rad}$ per scan.

We have obtained 24 different scans moving the robot around the lab and rotating it an angle $\beta \leq 0.2 \text{ rad}$. Data results from the scan matching are shown from two different points of view in Fig. 7a and in Fig 7b. Points A and B from Fig 7 correspond to start and ending points of the scan, respectively. The Scan Matching has been colored on the values of the Y axis (height) in order to make it more visually understandable. Results seem to be quite real and accurate.

After this 24 scans we started increasing the rotation between our scans over 0.2 rad ($\beta \geq 0.2 \text{ rad}$). Then we experienced some error on the applied ICP algorithm, as shown in Fig. 8. It can be seen how walls of our scan start to get deviated as we start increasing the angle of the movements. This could be due to the fact that the quality of alignment obtained by ICP depends heavily on choosing good pairs for corresponding points in the two cloud point [17]. Again, points A and B from Fig 8 correspond, respectively, to start and ending point of the complete scan.

Since ICP does not allow our robot enough freedom in its movements, we have performed a mapping implementation using odometry. To obtain the needed odometry information we have used the *differentialRobot* component that our frame-

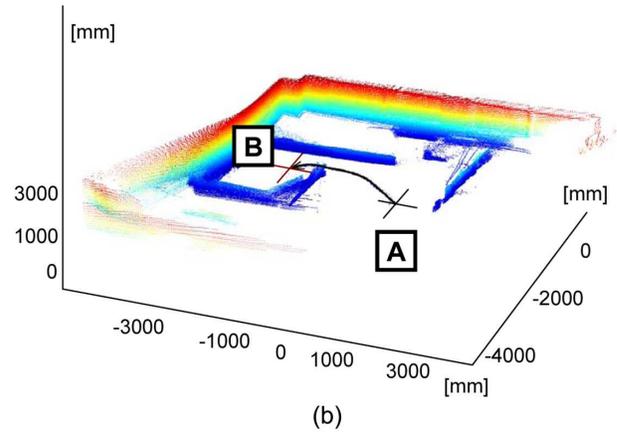
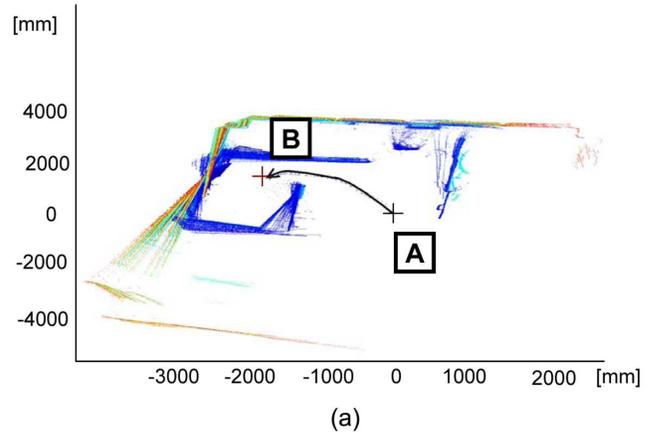


Fig. 8: ICP Scan Matching using 31 different scans and moving from A to B increasing the angle over $\beta \geq 0.2 \text{ rad}$.

work *Robocomp* provides (see (2) in Fig. 4). We have done 31 scans rotating an angle of $\beta = 0.2 \text{ rad}$ one of our Autonomous Mobile Robots (labeled A on Fig. 9). This movement specially confuses the ICP algorithm, and makes it almost impossible to make the match without using odometry. Therefore, making use of the obtained points, combined with the odometry, we have performed the matching showed on Fig. 9. As you can see results are much better than the ones we obtained with ICP. Still an accumulated error on the final scan is showed by the red square, labeled b in Fig. 9b. This, as further explained on Section V, could be solved using for the mapping, along with the 3D data, the texture information our system provides.

B. Novelty Detection and 3D Shape Retrieval based on Gaussian Mixture Models and superquadrics

The second test we have performed on our system is in the field of novelty detection and 3D shape retrieval. We have used the Gaussian Mixture Model for novelty detection and superquadrics for the 3D shape retrieval [18]. There are some main steps in the algorithm: the multi-scale sampling to reduce computation burden; change detection based on Earth's Mover Distance over the point cloud selections of

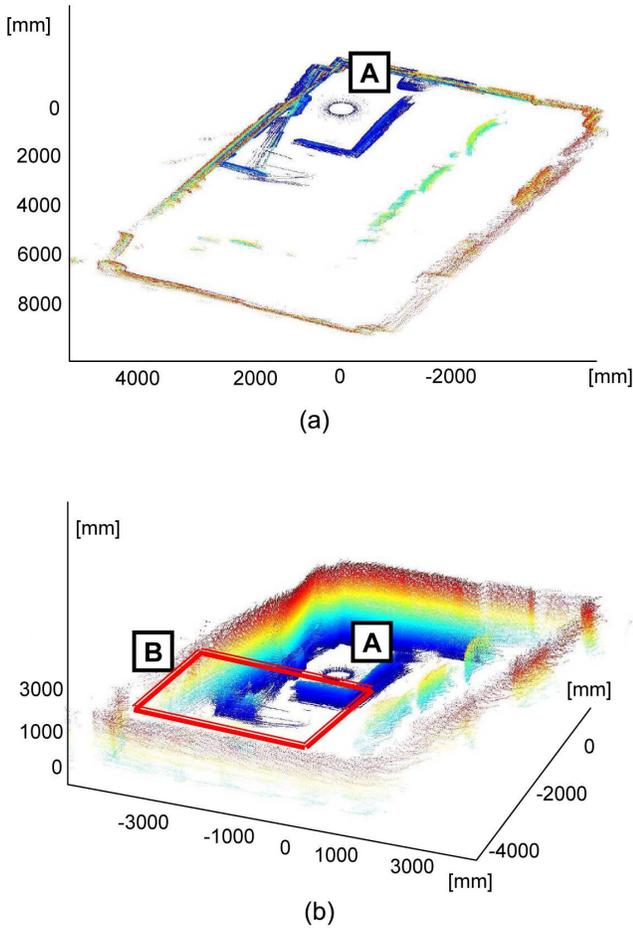


Fig. 9: Data Obtained from a scan in our lab using odometry. Point A shows where the robot has rotated and the red square the accumulated odometry error.

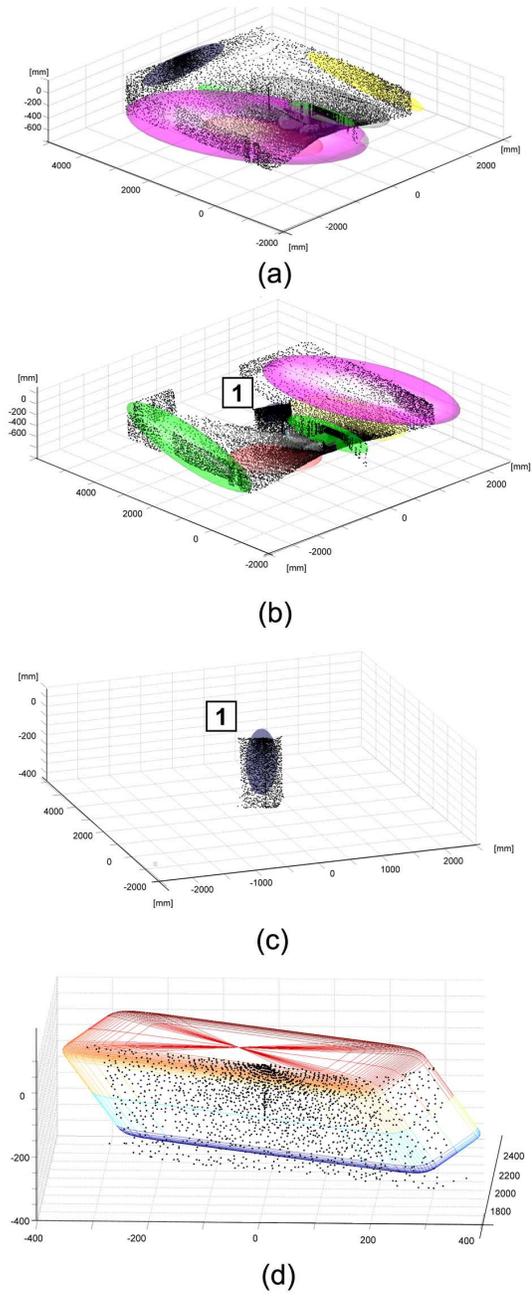


Fig. 10: (a) Simplified point cloud and associated Gaussians of empty room (b) Simplified point cloud and associated Gaussians of room with a novelty (c) Selected novelty on the second scan and associated point cloud (d) Retrieved superquadric from the novelty selected point cloud.

the Gaussian Mixture Model and a 3D shape retrieval of the detected changes using superquadrics.

To perform our experiment we have used our system to scan an empty room in our lab. Then a box has been added to the scene in order to introduce a novelty on the environment. Fig. 10a shows the simplified point cloud (in black) and the correspondent Gaussians associated to the big concentration of points on the empty room, Fig. 10b shows the scene containing the novelty, labeled as (1). Then Gaussians from the first scan are compared to those on the second one and matched using the Earth Mover’s Distance EMD. The novelty usually shows up as an unmatched Gaussian as it is the only thing that wasn’t there before (labeled (1) in Fig. 10c).

After selecting the correspondent novelty we have retrieved a superquadric and placed it on the same place as the box was. The idea of this is to retrieve the 3D shape of the object that was representing the obtained point cloud, in this case a box. Fig. 10d shows a superquadric corresponding to the novelty detected by the previous steps of the algorithm.

This shape retrieval is important in order to provide the mobile robot with a geometric idea of the objects it is facing.

The GMM and superquadrics approach tested here seems quite interesting and challenging. However in our opinion further works are needed since it seems still highly dependable on thresholds and too slow with large datasets, at least if it wants to be used for real-time processing.

V. CONCLUSIONS AND FUTURE WORK

This paper has presented the construction of a 3D sensing system for Autonomous Mobile Robots consisting in a Camera, a 3D LRF and an embedded system. It is intended to be small and power efficient without giving up performance. We have tested our system retrieved data with two different algorithms obtaining promising results. Certainly the step motor combined with the 2D LRF constitutes a solid alternative to those expensive commercial solutions. Data is acquired accurately and fast enough while keeping the low-cost and autonomous requirements. Also, texture information is properly attached to the points coming from the LRF in order to get more information for data processing algorithms. The public interface provided by the communication middleware makes the whole system become a real hardware component, accessible externally through the network. The API offered by the laser component, provides methods to actively scan the world with variable precision.

For the mapping, scan matching using ICP and odometry has been performed. Results are good but still depend on the point cloud provided for ICP, making the algorithm sensible to high changes between scans. Making use of texture information [19] or even making it a real-time mapping with small changes between consecutive scans could constitute a good upgrade for this experiment. In the novelty detection and 3D shape retrieval field the used algorithms (GMM and superquadrics) seemed promising. However, we found some trouble detecting certain point clouds, probably because these algorithms seem highly sensible to thresholds. They, also where very intensive in CPU and memory. Again adding texture information to the algorithm might be a choice, although this could make the application not suitable for real-time use due to a high CPU load. The 2D LRF makes scans at a speed up to 25ms, although our system still retrieves data at a speed of 50ms. This is caused by the embedded processor's CPU whose limited performance still constitutes a bottleneck. New and more powerful ARM processors with multicore architecture have already been announced by Texas Instruments. They are expected for the next few months and will probably solve this problem.

Finally, texture information is attached to the laser through the manual calibration process from section III-B2, another interesting improvement could be to develop an automatic way of finding these camera-laser data correspondences.

ACKNOWLEDGMENTS

This work has partly been supported by grants PRI09A037 and PDT09A044, from the Ministry of Economy, Trade

and Innovation of the Extremaduran Government, by grant TSI-020301-2009-27 from the Spanish Ministry of Industry, Tourism and Commerce, by grant IPT-430000-2010-002 from the Spanish Ministry of Science of Innovation and by FEDER funds from the European Community.

REFERENCES

- [1] N. Ayache and F. Lustman, "Trinocular stereo vision for robotics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 1, pp. 7385, 1991.
- [2] J. Meguro, J. Takiguchi, Y. Amano, and T. Hashizume, "3D reconstruction using multibaseline omnidirectional motion stereo based on gps/dead-reckoning compound navigation system," *Int. Journal of Robotics Research*, vol. 26, pp. 625636, 2007.
- [3] H. P. Moravec, "Robot spatial perception by stereoscopic vision and 3D evidence grids," *CMU Robotics Institute Technical Report CMU-RI-TR-96-34*, 1996.
- [4] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IEEE Workshop on Stereo and Multi-Baseline Vision*, 2001.
- [5] L.J. Manso, P. Bachiller, P. Bustos, P. Nuñez, R. Cintas and L. Calderita. "RoboComp: a Tool-based Robotics Framework". In *Proc. of Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAP)*. Pages 251-262. 2010.
- [6] Trucco, Emanuele and Verri, Alessandro., "Introductory techniques for 3D Computer Vision", Prentice-Hall Inc, 1998.
- [7] Wulf, Oliver and Wagner, Bernardo, "Fast 3D Scanning Methods for Laser Measurement Systems", *Proceedings of the International Conference on Control Systems and Computer Science*, 2003.
- [8] Davis, J.; Chen, X.; , "A laser range scanner designed for minimum calibration complexity," *Proceedings. Third International Conference on 3-D Digital Imaging and Modeling*, 2001, vol., no., pp.91-98, 2001
- [9] Vidal-Calleja, T. and Davison, A.J. and Andrade-Cetto, J. and Murray, D.W., "Active control for single camera SLAM", *Proceedings IEEE International Conference on Robotics and Automation*, 2006.
- [10] Furukawa, Yasutaka; Curless, Brian; Seitz, Steven M.; Szeliski, Richard; , "Reconstructing building interiors from images," *IEEE 12th International Conference on Computer Vision*, 2009, pp.80-87.
- [11] Aliakbarpour, H.; Dias, J.; , "IMU-Aided 3D Reconstruction Based on Multiple Virtual Planes," *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2010, pp.474-479.
- [12] Bosse, Michael; Zlot, Robert; , "Continuous 3D scan-matching with a spinning 2D laser," *International Conference on Robotics and Automation*, 2009. ICRA '09. pp.4312-4319, 12-17.
- [13] Ryde, J.; Huosheng Hu; , "3D Laser range scanner with hemispherical field of view for robot navigation," *International Conference on Advanced Intelligent Mechatronics*, 2008. AIM 2008. IEEE/ASME, vol., no., pp.891-896.
- [14] Hartmut Surmann and Kai Lingemann and Andreas Nchter and Joachim Hertzberg, "A 3D laser range finder for autonomous mobile robots," *Proceedings of the 32nd International Symposium on Robotics*, 2001 pp. 153-158.
- [15] Ryde, J.; , "An inexpensive 3D scanner for indoor mobile robots," *International Conference on Intelligent Robots and Systems, (IROS) 2009*. IEEE/RSJ, pp.5185-5190.
- [16] Chen, Y.; Medioni, G.; , "Object modeling by registration of multiple range images," *Proceedings IEEE International Conference on Robotics and Automation*, 1991. pp.2724-2729 vol.3.
- [17] Gelfand, N.; Ikemoto, L.; Rusinkiewicz, S.; Levoy, M.; , "Geometrically stable sampling for the ICP algorithm," *Proceedings. Fourth International Conference on 3-D Digital Imaging and Modeling*, 2003. 260- 267.
- [18] Drews, P.; Nuñez, P.; Rocha, R.; Campos, M.; Dias, J.; , "Novelty detection and 3D shape retrieval using superquadrics and multi-scale sampling for autonomous mobile robots," *International Conference on Robotics and Automation (ICRA)*, 2010 IEEE, pp.3635-3640.
- [19] Ji Hoon Joung; Kwang Ho An; Jung Won Kang; Myung Jin Chung; Wonpil Yu; "3D environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3D laser range finder," *International Conference on Intelligent Robots and Systems*, 2009. IEEE/RSJ , pp.3082-3088.
- [20] The primesensor Technology, <http://www.primesense.com/>