

Generation of laser-quality 2D navigation maps from RGB-D sensors

Federico Nardi^[0000-0002-9275-7152], María T. Lázaro^[0000-0002-7742-2442], Luca Iocchi^[0000-0001-9057-8946], and Giorgio Grisetti^[0000-0002-8038-9989]

Dept. of Computer, Control and Management Engineering
Sapienza University of Rome, Italy
{fnardi, mtlazaro, iocchi, grisetti}@diag.uniroma1.it

Abstract. The use of RGB-D cameras has become an affordable solution for robot mapping and navigation in contrast to expensive 2D laser range finders. Although these sensors provide richer information about the 3D environment, most successful mapping and navigation techniques for mobile robots have been developed considering a 2D planar environment. In this paper, we present our system for 2D navigation using RGB-D sensors. The key feature of our system is the extraction of 2D laser scans out of the 3D point cloud provided by the camera that can be later used by common mapping or localization approaches. Along with the real experiments we raise the question “*how far can we go with the use of RGB-D sensors for 2D navigation?*” and we analyze performance and limitations of the system compared to accurate, yet expensive, laser-based systems.

Keywords: RGB-D sensors, 2D mapping, educational robots

1 Introduction

Robot mapping and navigation in indoor environments are fundamental abilities in many different application scenarios, including home service robots (e.g., cleaning robots) and educational robots. In particular, several competitions in RoboCup require such abilities, especially within the RoboCup@Home and RoboCupIndustrial domains, and other junior competitions would benefit from these functionalities.

Two major components must be developed: a mapping system, able to generate a map of the environment, and a navigation system, able to plan and execute paths in this environment. The quality of these processes depends on the combination of the sensors and the software used. High-quality sensors provide better quality of data that guarantees accuracy and robustness of the mapping and navigation processes. However, high-quality sensors are also expensive, thus increasing the budget to build and maintain robots and the difficulty of teams in participating to robot competitions. Consequently, most applications involving home service robots do not use a proper mapping, localization and navigation system, significantly limiting their functionalities.

Examples of high-quality sensors for robot mapping and navigation are 2D laser-range finders (LRF). Several models exist with different technical specifications (e.g., Hokuyo UTM and URG family, SICK LMS and TIM family). 2D LRFs provide for very precise and accurate measurements and have been extensively used for mapping

and navigation in many robotic applications. However, they also have two drawbacks: 1) being planar sensors they are not able to deal with obstacles that are not on the same plane of the laser, 2) they are more expensive than camera-based solutions.

A second type of sensors are RGB-D cameras. Also in this case there is a variety of models with different specifications (e.g., ASUS Xtion PRO LIVE , Microsoft Kinect). RGB-D cameras provide for information about the environment that are not limited to a planar surface and are less expensive than LRFs, but they have the following disadvantages: 1) lower resolution, precision and accuracy of the acquired data, 2) limited field of view and range.

In this paper, we want to fill the gap between LRF and RGB-D technology, by presenting an effective mapping system that is based on RGB-D cameras. The proposed system provides results that are comparable with maps generated by a LRF. With the proposed method, it is thus possible to map and navigate in an environment with performance that is comparable with LRF technology at a reduced cost and with the advantage of using a 3D sensor for obstacle avoidance. This method can be useful on many kinds of robots and in many applications and RoboCup competitions, since it enables the possibility of performing complex tasks with a low-cost robot. For example, RoboCup@Home Education¹ is an initiative aiming at providing support to teams to set-up RoboCup@Home teams with low-cost hardware. In this initiative, robots are equipped only with a single RGB-D sensor. Another example is given by the Soft-Bank Pepper robot used in RoboCup@Home Social Standard Platform League that is equipped with a laser device providing only a few laser scans and, therefore, navigation tasks highly depend on its RGB-D sensor.

More specifically, the contributions of this paper are: 1) a method to generate 2D laser measurements from the information gathered by an RGB-D sensor based on a projection procedure that allows to extract a scan-line at an arbitrary laser pose (in contrast to ROS² tool *depthimage_to_laserscan*³ which generates the laser measurement from a fixed pose), making it suitable also for tilted or moving cameras, 2) the adoption of a robust SLAM system to deal with lower performance – namely precision and field-of-view – of RGB-D sensors compared to LRFs, and 3) the evaluation of navigation systems of low-cost robots when mapping and navigation are performed with only an RGB-D sensor.

We validate our approach in two sets of experiments. In the first set, we compare the quality of the maps produced with RGB-D sensors with respect to those obtained with an expensive and high accurate LRF sensor, while in the second set we evaluate navigation capabilities of a robot using an RGB-D sensor and a motion model that are different from the ones used for mapping.

2 Related Work

An efficient way to represent an indoor environment for navigation is to use occupancy grids, first introduced in the work of Moravec and Elfes [12]. This technique consists

¹ www.robocupathomeedu.org

² <http://www.ros.org/>

³ wiki.ros.org/depthimage_to_laserscan

in integrating the range measurements acquired by the robot into a common frame to represent the obstacles in the scene. The use of laser scanners for building such a map allows to obtain high accuracy in the measurements but implies, also, a significant economic expense. Since accurate digital cameras became available at reasonable costs, there have been different contributions in the effort of extracting range measurements from cameras in order to avoid the expense of laser scanners.

Murray and Jennings [13] introduced the first occupancy-grid mapping system that uses a visual sensor to collect dense depth information. They manage to produce depth images with a trinocular camera, which yields accurate results but, of course, requires a specialized rig and a meticulous calibration. Sim and Little [15] use stereo-vision to perform RaoBlackwellised Particle Filter (RBPF) visual SLAM. Although their method can be categorized as feature-based SLAM, the authors describe how to infer the corresponding occupancy grids using a maximum likelihood estimate.

Gastshore et al. [5] were the first to explore the possibility of building occupancy grids with a single camera. In their work, each grid cell represents a vertical line feature and a voting scheme is used to propose the occupancy of each grid cell. The authors, however, don't consider problem of localization with this method. To really obtain range measurements from a monocular camera Choi and Oh [3] introduce the visual sonar technique, which consists of projecting in space virtual rays and then detecting where these rays collide with obstacles in 3D space. Asmar and Samir [1] extend the previous approach to deal with variable lighting conditions by using g High Dynamic Range (HDR) images and propose a different segmentation technique to have a better extraction of the ground features.

Since the appearance of the first RGB-D cameras in 2010, robotic researchers have been attracted by the potentialities of this type of sensors for mobile robots navigation. This is because they provide depth information for each image pixel and they are available at a reasonable price. In this context, one of the first works to use this sensor for navigation has been made by Oliver et al. [14], in which they simulate the presence of a laser by extracting a 2D scan from the depth image. Our approach is similar to that of Oliver et al., in addition, we use reliable calibration procedures on our system to obtain precise intrinsic and extrinsic camera parameters. This allows us to obtain accurate maps, which can be used later for navigation as we show in Sec. 4.

3 Our Approach

The 2D mapping procedure from RGB-D sensors proposed in this paper can be summarized in the following steps. Firstly, a data acquisition phase is performed while steering the robot in the environment to be mapped, in which the sequence of RGB and Depth images acquired by an RGB-D camera together with the robot odometry are recorded.

Next, each depth image is used to reconstruct the local 3D scene as a point cloud from which a virtual scan is extracted to simulate the presence of a laser device.

The RGB-D scans together with the robot odometry are used as input to our robust laser-based graph SLAM approach to estimate the robot trajectory. The result of this process provides the estimated global robot coordinates from which each depth image (RGB-D scan) was acquired.

Finally, the map of the environment is obtained by fusing the generated scans into an occupancy grid based on the pose-graph computed at the previous step. In the remainder, we describe in detail each component of the system.

3.1 Extraction of scans from depth images

The first step of our approach is to generate range measurements for building the 2D map. As mentioned in Sec. 1, ROS provides the package *depthimage_to_laserscan* to perform this operation, which is done by extracting the set of pixels belonging to the middle row of the depth image and back-projecting them in 3D.

It is relevant to notice that this approach is effective only under the assumption of a camera parallel to the ground floor, see Figure 1. Instead, our approach works by first transforming the depth cloud in the virtual laser frame and consequently extracting scan points. This allows a better flexibility, since the technique can be applied also to a tilted camera, as it is that of the Pepper, and it is possible to extract points whose corresponding pixels fall outside the middle row of the image (see below).

The proposed procedure to generate virtual laser scans is the following. Each pixel $\mathbf{u} = (u_i, u_j) \in \mathbb{R}^2$ with depth value $d_{ij} \in \mathbb{R}$ of the current depth image is back-projected into the corresponding 3D point $\mathbf{p} = (p_x, p_y, p_z) \in \mathbb{R}^3$ using the camera calibration matrix \mathbf{K} [7]:

$$\mathbf{p}^c = \mathbf{K}^{-1} \cdot [u_j d_{ij} \quad u_i d_{ij} \quad d_{ij}]^T \quad (1)$$

where superscript \mathbf{p}^c denotes the point is represented with respect to the camera frame.

Let $\mathcal{P}^c = \{\mathbf{p}_k^c\}$ be the point cloud containing the collection of k transformed points in the camera frame. Given \mathbf{T}_c^r , the pose of the camera with respect to the robot frame and using \mathbf{T}_v^r as the pose of the virtual laser frame from which we want to generate an RGB-D scan, it is possible to obtain the point cloud $\mathcal{P}^v = \{\mathbf{p}_k^v\}$ in the laser frame by applying the following transformation to each point:

$$\mathbf{T}_c^v = (\mathbf{T}_v^r)^{-1} \cdot \mathbf{T}_c^r \quad (2)$$

$$\mathcal{P}^v = \mathbf{T}_c^v \cdot \mathcal{P}^c. \quad (3)$$

Finally, we set the 2D scan to lie in the XY plane of the laser reference frame. To this end, we generate the RGB-D scan \mathbf{s}_v by considering only the points of \mathcal{P}^v whose height is inside a pre-defined threshold ε :

$$\mathbf{s}_v = \{\mathbf{p}^v \in \mathcal{P}^v \text{ such that } |p_z| \leq \varepsilon\}, \quad (4)$$

where ε is typically chosen to be close to zero in order to allow for some tolerance.

3.2 Estimation of the robot trajectory

In order to generate a consistent trajectory of the robot we use our novel graph-based SLAM system [9]. Since describing mathematical details of our SLAM system is out of the scope of this paper, we briefly describe the main pipeline and specific features

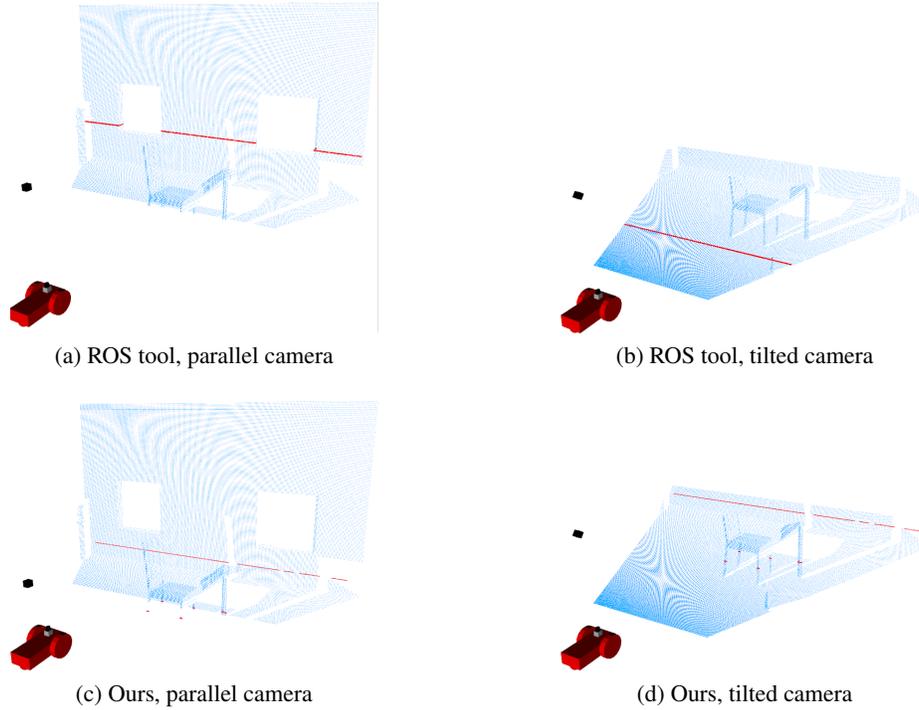


Fig. 1: Comparison of the extraction procedure with the ROS *depthimage_to_laserscan* tool and our method. The RGB-D camera is represented by the black box on top of the robot, the virtual laser is represented by the 3D model of an Hokuyo sensor.

that make it suitable for mapping 2D environments using scans extracted from RGB-D sensors. The input of the system consists of a stream of laser scans and the robot odometry and the output is the estimation of the trajectory followed by the robot.

In such system, the map of the environment is represented as a graph whose vertices contains robot positions together with a measurement (laser scans in this case) gathered from that position and the edges connecting a pair of vertices represents spatial constraints between them. Then, the goal of the SLAM system is to obtain a global configuration of the vertices that better explains the constraints. This is done by formulating a least-squares minimization problem that can be solved by using optimization methods like Gauss-Newton or Levenberg-Marquardt.

The pipeline of the SLAM system used in this work approaches the following aspects:

laser tracking Is the problem of tracking the position of the laser between consecutive timestamps. This is done by implementing an Iterative Closest Point (ICP) algorithm, which provides the transform that minimizes the reprojection error of the points of the current scan with respect to the previous one.

local-map generation and management Is the problem of generating a consistent local view of the environment out of a small chunk of trajectory. To this end, the scans registered during the laser tracking process are continuously fused into a single 2D point cloud producing a refined representation of the environment. Local map maintenance is done by removing points further than a certain distance from the current tracked pose. When the robot moves for a certain distance, the current local map is stored and associated to a vertex in the graph. The use of local maps in contrast to the use of single laser scans greatly improves the reconstruction of the local surroundings of the robot specially in the case of the use of RGB-D sensors due to their limited field of view.

relocalization or loop closing Is the task of determining if the current local map captures a portion of the environment already seen in the past. To this end our SLAM approach first retrieves which local maps from those already present in the graph are most similar to the current one by selecting the vertices in covariance range with respect to the current one, then it validates the potential matches by performing local map registration. This generates a set of possible *closures* that are finally introduced in the graph as edges connecting non-consecutive vertices after an inlier verification process using a voting scheme described in [11].

global optimization Relocalization events might introduce “jumps” in the estimated trajectory. The global optimization module aims at distributing the estimation error evenly between the vertices. This is done in the SLAM system by solving a pose-graph [6] by means of the **g2o** optimizer [8]. The core idea is to determine the positions of the nodes that better explain the relative transformations arising from laser-tracking and relocalization.

At the end of a run, this mapping process returns an estimate of the robot trajectory $\{\mathbf{x}_{1:t}^r\}$ from which the laser scans were generated. The whole described SLAM process accounts for the transformation \mathbf{T}_i^r to represent the laser scans into the robot reference frame.

3.3 Occupancy grid for navigation

Once the trajectory is provided by the SLAM system, it is possible to reconstruct the most likely map using the gathered measurements. If we want our robot to navigate safely in the environment, i.e. without hitting obstacles, we have to provide it with the information about free and occupied space. For this purpose, we represent the environment with an *occupancy grid* [12], where each cell contains the probability of being occupied by an obstacle, thus containing values ranging from 0 (free) to 1 (occupied).

This method goes under the name of *occupancy grid mapping* [16], and in the remainder of the paragraph we report a sketch of the procedure. The inputs are the RGB-D scans $\{\mathbf{s}_{v_{1:t}}\}$ and the estimated robot poses $\{\mathbf{x}_{1:t}^r\}$. For each range of the single laser scan, a ray is casted with a straight-line grid traversal algorithm [2] to find the traversed cells. Each cell of the grid stores two values: the number of *hits* and the number of *misses*. If a ray passes through a cell we increment its misses count, while for the end-point of the ray we increment the hits count of the cell where it falls.

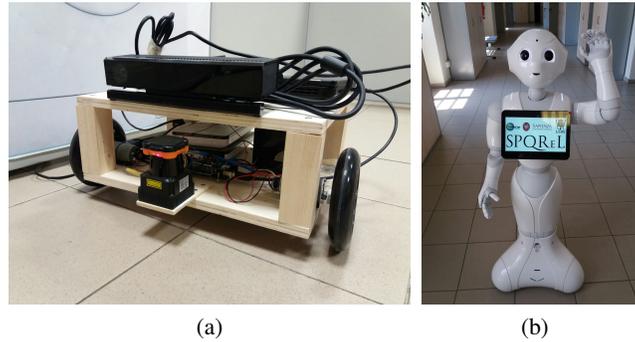


Fig. 2: Robots and sensors configuration used for the experiments. a) MARRtino robot. b) Soft-Bank's Robotics Pepper.

Finally, the probability of each cell of being occupied is computed as $\frac{hits}{hits+misses}$. The selection of the appropriate threshold value allows us to discriminate between three types of cells:

- Unexplored: $hits = 0, misses = 0$
- Occupied: $\frac{hits}{hits+misses} > threshold$
- Free: $\frac{hits}{hits+misses} \leq threshold$

which is used to obtain a thresholded grid map compatible with the most commonly used localization and path planning algorithms for navigation purposes.

4 Experiments

The main goal of this work is to produce usable 2D navigation maps from RGB-D data. Our key claim is that the proposed mapping system, using data from an affordable RGB-D sensor, can provide comparable quantitative and qualitative results to those that we could obtain with a more accurate and expensive laser range finder. Therefore, in this section, we present real experiments in which our approach is used to create maps of different indoor environments which can later be used to perform common navigation tasks such as localization or path planning, not necessarily with the same robot used to build the map.

The first set of experiments are intended to evaluate the quality of the maps produced by our proposed system. To this end, we used a self-built robot (Fig. 2a) based on the open source and open hardware robotic platform MARRtino Robot⁴, and equipped it with a Microsoft Kinect for Windows V2 RGB-D sensor and an Hokuyo UTM-30LX laser range finder. The difference in hardware specifications between both sensors is highlighted in Table 1. We implemented the components of our system in C++ as ROS nodes, publicly available as open source^{5, 6}.

⁴ <http://www.dis.uniroma1.it/~spqr/MARRtino/>

⁵ https://gitlab.com/srrg-software/srrg_depth2laser_ros

⁶ https://gitlab.com/srrg-software/srrg_mapper2d_ros

	Hokuyo UTM-30LX	Microsoft Kinect V2
Field of view	270°	70°
Min range	0.1m	0.5m
Max range	30m	4.5m
Frames per second (FPS)	40Hz	30Hz
Power consumption	< 8W	2.25W
Resolution at 2m	1mm	5mm

Table 1: Comparison of the technical specification of the two sensors.

As a prior step before performing the experiments, we calibrated the robot to obtain the extrinsic sensor parameters \mathbf{T}_c^r and \mathbf{T}_l^r using the unsupervised calibration procedure explained in [4].

We tested our approach in two different indoor environments: a domestic environment of $60m^2$ and a corridor of our department at the Sapienza University of Rome, hereinafter denoted as \mathcal{M}_{home} and \mathcal{M}_{uni} respectively. The experimentation and validation is performed in two steps. First, we record a dataset containing the laser scans and the RGB-D data gathered by both on-board sensors together with the odometry provided by the platform while manually driving the robot around the test environments. This dataset is used to produce a map of the environment using our system. Then, once each map is constructed, we record a second dataset where the robot follows a different trajectory than the one used for mapping. We use this second dataset to validate localization with respect to the previously created maps using both sensors.

In order to validate our approach, we compare the map \mathcal{M}_{our} , built with virtual scans $\{s_{v_{1:t}}\}$, with respect to the map \mathcal{M}_{ref} , built with the real scans, denoted as $\{s_{r_{1:t}}\}$ and collected from the Hokuyo laser. Results are shown in Fig. 3. We also show that using available ROS tools, i.e. *depthimage_to_laserscan* + *gmapping*, in these scenarios does not provide a robust solution, since the output maps cannot be used for navigation tasks.

To obtain a quantitative comparison, we compute the percentage difference between the two corresponding images with the following formula:

$$\Delta = 100 \sum_{k=1}^K \left| \frac{u_{our_k} - u_{ref_k}}{K} \right| \quad (5)$$

where u_{ref_k} and u_{our_k} are the k^{th} pixel values of \mathcal{M}_{ref} and \mathcal{M}_{our} respectively. From the experiments we obtain that the difference of \mathcal{M}_{ref} is 1.86% with \mathcal{M}_{home} and 1.27% with \mathcal{M}_{uni} . This shows that if the system is properly calibrated then the RGB-D sensor can return a map that is almost undistinguishable from the one provided by the laser. The main difference between these sensors is in the angular extension of the measurement since the depth camera has a field of view that is less than one third of the laser scanner one. As a direct consequence, the map generated by the laser has some details that could not be detected by the Kinect due to its restricted angle of view, as shown in detail in Fig. 4.

The second set of experiments aim at evaluating the usability of the maps generated from the previous experiments for navigation tasks by robots with different features

		Home University	
Translational error (m)	Mean	0.034	0.075
	Max	0.123	0.115
	Median	0.025	0.085
Rotational error (rad)	Mean	0.017	0.010
	Max	0.218	0.036
	Median	0.013	0.009

Table 2: Translational and rotational localization errors with respect to the robot positions given by a localization system using the Hokuyo data ($s_{real} + \mathcal{M}_{ref}$)

(Fig. 2). Thus, we choose to test the particle filter based localization system in [10] to estimate the robot motion in different scenarios. To support our claim, we try to localize the MARRtino robot on \mathcal{M}_{our} by using the scans generated from the RGB-D sensor and we compare the resulting trajectories with the ones estimated by the localizer on \mathcal{M}_{ref} using the laser scans, Fig. 5. That is, we measure the translational and rotational error for each robot pose between the reference trajectory and the one obtained with the RGB-D scans. As it is shown in Fig. 6, in some points of the trajectory the localization with the depth camera is less precise than the reference one. To reduce this difference we have accurately calibrated the odometry parameters of the mobile platform, as explained in [4], in order to have a better initial guess of the robot motion. Results are reported in Table 2, showing overall localization errors of $\sim 5\text{cm}$ (translational) and $\sim 0.02\text{rad}$ (rotational).

Furthermore, we test University \mathcal{M}_{our} map during a navigation task with the Pepper robot. Pepper’s original laser consists of 45 points with a usable range of $\sim 3\text{m}$. As it’s shown in Figure 6c, the trajectory reported by the localization system is comparable with the ones executed by robots with more accurate sensors.

5 Conclusions

In this paper, we presented a novel approach to generate accurate 2D maps by using RGB-D cameras. We implemented and evaluated our approach on different datasets acquired in different application scenarios where the use of available packages would not give the same quality of the results. Quantitative experimental results show the effectiveness of the proposed approach, that is able to generate maps whose quality is comparable with the ones generated by laser sensors.

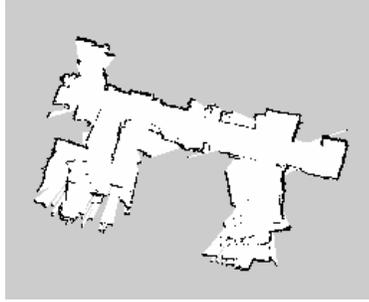
More specifically, we showed that self-localization performance of a robot in different environments are substantially the same when using an RGB-D camera or a laser range-finder. Therefore, this method enables the replacement of laser-based navigation with RGB-D navigation in many applications, yielding a lower cost and improved 3D information for obstacle avoidance.

Despite the results show high effectiveness of the proposed method in the considered situations, we would like to investigate other interesting situations of mapping and navigation for robots equipped with low-cost sensors. Moreover, we want to carry out additional experiments to evaluate in a quantitative way if the performance of depth

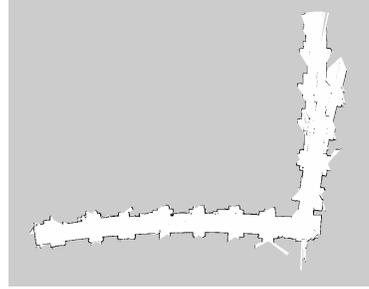
sensors on typical navigation tasks (e.g., entering a small passages, navigation speed, etc.) are comparable with those of laser-based solutions.

References

1. Asmar, D., Shaker, S.: 2d occupancy-grid slam of structured indoor environments using a single camera. *International Journal of Mechatronics and Automation* **2**(2), 112–124 (2012)
2. Bresenham, J.E.: Algorithm for computer control of a digital plotter. *IBM Systems journal* **4**(1), 25–30 (1965)
3. Choi, Y.H., Oh, S.Y.: Map building through pseudo dense scan matching using visual sonar data. *Auton. Robots* **23**(4), 293–304 (Nov 2007)
4. Cicco, M.D., Corte, B.D., Grisetti, G.: Unsupervised calibration of wheeled mobile platforms. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*. pp. 4328–4334 (May 2016). <https://doi.org/10.1109/ICRA.2016.7487631>, https://gitlab.com/srrg-software/srrg_nw_calibration_ros
5. Gartshore, R., Aguado, A., Galambos, C.: Incremental map building using an occupancy grid for an autonomous monocular robot. In: *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th Int. Conf on. vol. 2*, pp. 613–618. IEEE (2002)
6. Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W.: A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine* **2**(4), 31–43 (2010). <https://doi.org/10.1109/MITS.2010.939925>
7. Hartley, R., Zisserman, A.: *Multiple view geometry in computer vision*. Cambridge university press (2003)
8. Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g2o: A general framework for graph optimization. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*. pp. 3607–3613 (May 2011). <https://doi.org/10.1109/ICRA.2011.5979949>
9. Lázaro, M.T., Capobianco, R., Grisetti, G.: Efficient long-term mapping in dynamic environments. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Madrid, Spain (Oct 1-5 2018)
10. Lázaro, M.T., Grisetti, G., Iocchi, L., Fentanes, J.P., Hanheide, M.: A lightweight navigation system for mobile robots. In: *ROBOT 2017: Third Iberian Robotics Conference* (2017)
11. Lázaro, M.T., Paz, L.M., Piniés, P., Castellanos, J.A., Grisetti, G.: Multi-robot slam using condensed measurements. In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. Tokyo Big Sight, Japan (Nov 3-8 2013). <https://doi.org/10.1109/IROS.2013.6696483>
12. Moravec, H., Elfes, A.: High resolution maps from wide angle sonar. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*. vol. 2, pp. 116–121. IEEE (1985)
13. Murray, D., Jennings, C.: Stereo vision based mapping and navigation for mobile robots. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)* (1997)
14. Oliver, A., Kang, S., Wünsche, B.C., MacDonald, B.: Using the kinect as a navigation sensor for mobile robotics. In: *Proceedings of the 27th Conf. on Image and Vision Computing New Zealand*. pp. 509–514. IVCNZ '12, ACM, New York, NY, USA (2012)
15. Sim, R., Little, J.J.: Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters. In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. pp. 2082–2089. IEEE (2006)
16. Thrun, S., Burgard, W., Fox, D.: *Probabilistic robotics*. MIT press (2005)



(a) Home, ROS tools



(b) University, ROS tools

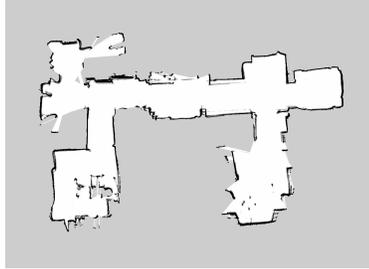
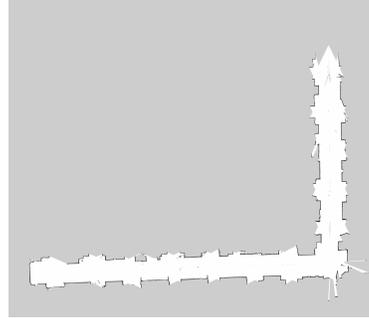
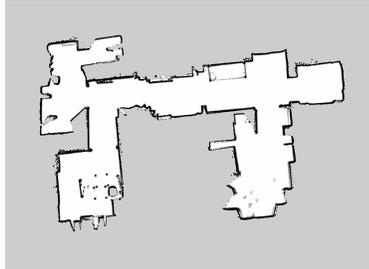
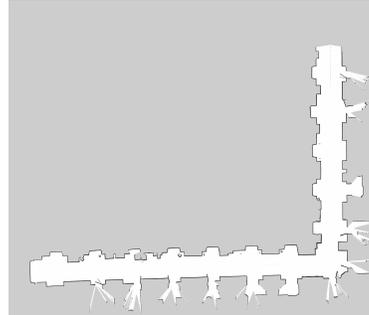
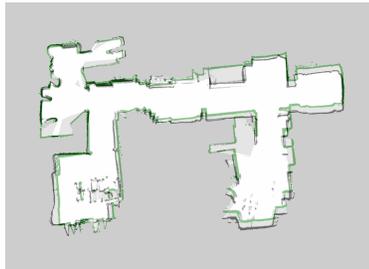
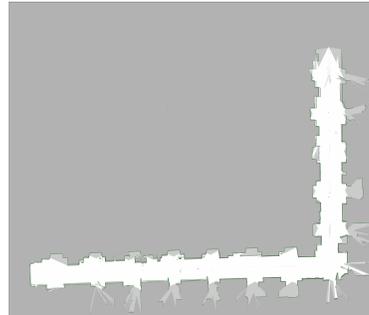
(c) Home, \mathcal{M}_{our} (d) University, \mathcal{M}_{our} (e) Home, \mathcal{M}_{ref} (f) University, \mathcal{M}_{ref} (g) $\mathcal{M}_{our} - \mathcal{M}_{ref}$ (h) $\mathcal{M}_{our} - \mathcal{M}_{ref}$

Fig. 3: Results of the first set of experiments. First row: maps built using available ROS tools. Second row: maps generated with RGB-D scans. Third row: maps generated with laser scans. Last row: superposition of \mathcal{M}_{ref} on \mathcal{M}_{our} , for a qualitative comparison we show in green the reference map on the map generated with our approach.

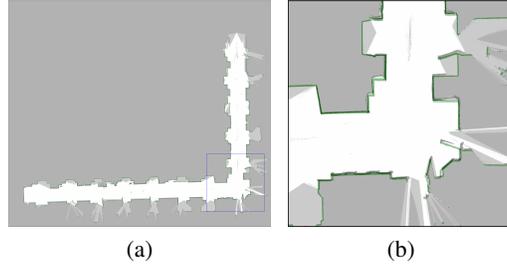


Fig. 4: (a) Corridor map. (b) Detail. The figure shows the \mathcal{M}_{ref} (in green) superimposed to \mathcal{M}_{our} .

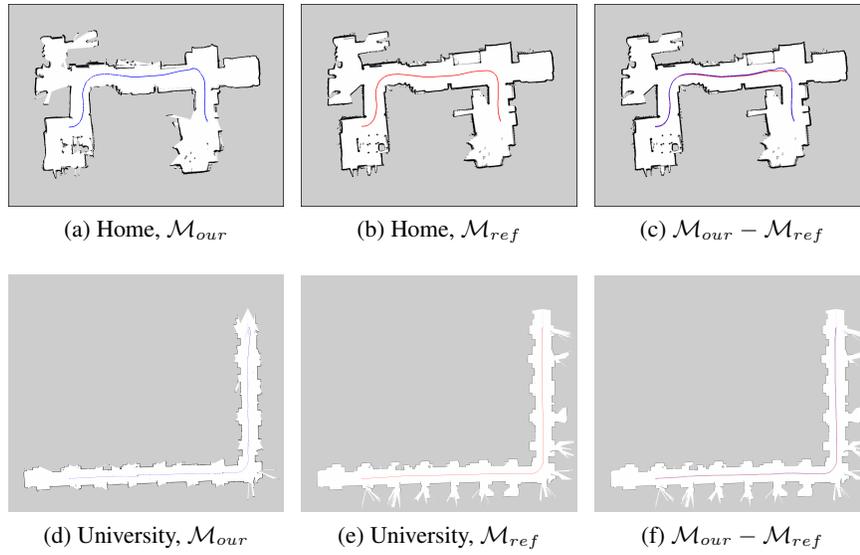


Fig. 5: Trajectories estimated by the localizer. Left column: $s_{virtual} + \mathcal{M}_{our}$. Center column: $s_{real} + \mathcal{M}_{ref}$. Right column: superposition of both trajectories.

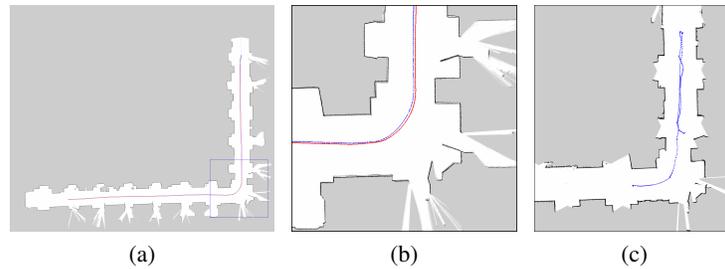


Fig. 6: (a - b) Estimated trajectories and zoomed area for the corridor map. (c) Trajectory followed by the Pepper robot during a navigation task at University in map \mathcal{M}_{our} .