Low cost robot for indoor cognitive disorder people orientation

S. Maldonado-Bascón, F.J. Acevedo-Rodríguez F. Montoya-Andúgar and P. Gil-Jiménez
Dpto. de Teoría de la Seal y Comunicaciones
Escuela Politecnica Superior
Universidad de Alcalá
Alcalá de Henares (Madrid)
http://agamenon.tsc.uah.es/Investigacion/gram/

***Abstract*—Temporal and spatial orientation problems arise with cognitive disorders. This paper describes an assistive low-cost robot, focusing on its navigation abilities based on ultrasound sensors and QR landmarks. Compared with other works, we propose the use of low-cost inaccurate sensing setup, which must be counteracted by proper design of signal processing algorithms. The results show that accurate enough occupancy map can be obtained with the developed platform.**

The assistive robot is intended to be used at disable attention centers, where non-technical-specialized caregivers are to be using it at their facilities, including the initial setup. For that reason, special attention has been paid in the ease of use. Although other functionalities are being included, the main application described in this paper is to give information to people with cognitive disorders, about their position and the path to follow to reach a specific location.

I. Introduction

Assistive robots have been an active field of research in the last decades [1], [2]. Nowadays the possibility to construct low cost robots is a reality and lots of applications can be developed using available technology. Applications in many occasions need special tunning for specific users. In many assistive applications, robots need to have the ability to move across the building or room where they are been used. When a robot, or any moving platform, is traversing a scenario, navigation and localization are the main issues. To help navigation, artificial landmarks have been considered long time ago [3], [4]. In our case, we introduced landmarks as the easiest way for the user to include essential information to the assistive application. For instance, the name or kind of a room with a landmark stuck to the frame of its entrance door. Once the landmarks are included we could also use them to refine map construction and navigation. We had also considered the use of hand-drawn sketched maps as in [5], but this method is difficult to be utilized for non-technical-specialized users at disable attention centers and such task will be time-consuming.

Cloud robotic has been considered in many approaches, but their requirements of internet connection can be a key rejection point in many centers. It is difficult to avoid such connection for some application, i.e., intensive computations or access to large databases. In [6] a set of cloud robots to support elderly and disable people are considered. One of those works, where a low cost robot is considered, can be found in [7]. However, influenced by our target applications and our experience in different disable attention centers, our approach tries to keep the robot computation and information tools apart from the Internet.

Our assistive system is based on a low cost platform intended to give information to disable users in an attention center about the position where they are and the way to follow to reach a specific location. We have included a screen, loudspeaker, microphone and camera apart from the sensors and motors. The work described in this paper allows the robot to obtain a reference map and navigate to a specific point in an unsupervised fashion. Target points have been tagged with QR codes. Furthermore, initial results on occupancy map are refined using particle correction [8], [9] and [10]. A simulated solution is presented in [11]. A gaussian model is used in [12] for the occupancy map construction to provide higher precision.

Our researh group have been working in developing many different technical aids under the solidarity program "Padrino Tecnológico" ($http://padrinotecnologico.org/$), from walkers to electric wheelchairs. This is an approach to use robots for assistive tasks. Orientation for people with cognitive disorder is stressing and helping to increase their autonomy is the first goal of this work although many others functionalities can be included using the same hardware.

II. Robot platform description

We have developed our own low cost robotic platform. A picture of this platform can be seen in Fig. 1 and main dimensions are depicted in Fig. 2. The wheels are 60mm in diameter. The low level control is performed with an Arduino Mega and, if necessary, a Raspberry Pi 3 can be attached. In the application described here, the Raspberry controls a RGB camera fixed in the center of the robot 100mm above the floor. Two speakers have been included in order to give sounds signals or voice message when an object is detected or a destination point reached. Voice message can also be used to inform about date and hour to give temporal information. Furthermore, the robot has an upper platform where other devices can be allocated and a 12V 7.2Ah battery is available in order to get a longer operation period if needed.

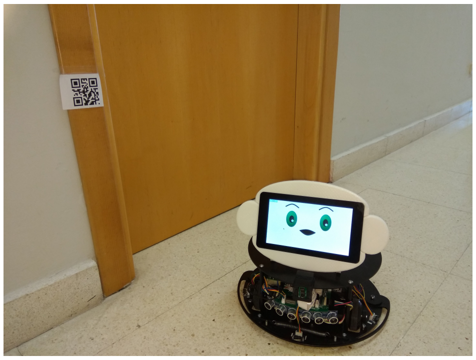


Fig. 1: A picture of the robot.

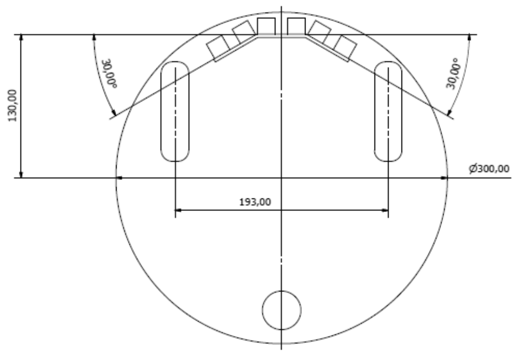


Fig. 2: Platform dimensions.

Focusing in the sensing setup, it is composed of 3 ultrasound sensors, 3 fall sensors and 3 bumpers, all of them located in the frontal side of the platform, as shown in Fig. 2. Those sensors are intended for navigation and map construction, although the fall and bumper sensors are used just in case of emergency. The movement is achieved with two 12V motors equipped with encoders with direct connection to the wheels. A 7 inchs touch-screen can be installed in the frontal part of the robot for debug or user interface applications.

III. Ultrasound range detection and map construction

As aforementioned, the main target of the design was to develop a low-cost robot to assist people with special needs. The limitations in price entail, among other things, the use of $HC-SR04$ ultrasonic sensors, instead of a much more expensive laser range device. It is well known that those ultrasound sensors do not give very accurate information. In our case, the sensor has a horizontal aperture of approximately 60 degrees, the measurements beyond 2 meters are quite inaccurate and it is very sensitive to object orientation. All these disadvantages must be overcome with proper signal conditioning and processing. The sensors are arranged horizontally, with a separation angle of 30 degrees between them, so the horizontal aperture of two adjacent sensors are overlapped 30 degrees, as shown in Fig. 3.

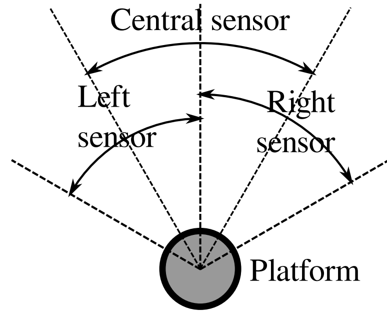


Fig. 3: Ultrasound sensor setup and horizontal apertures. Apertures are 30 degrees overlapped between adjacent sensors.

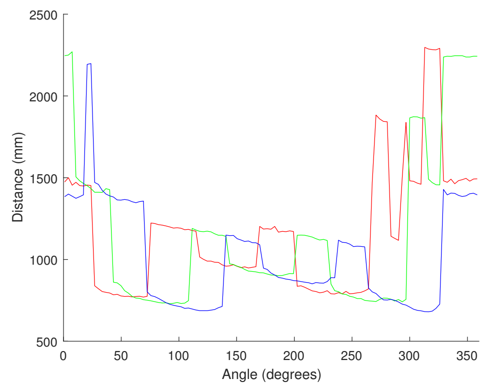


Fig. 4: Ultrasound readings for the scenario shown in Fig. 5 after one loop. (Red: left sensor, Green: central sensor, Blue: right sensor).

Most of the initial information for the map construction is acquired with circular movements of the platform around itself. Fig. 4 shows the sensors readings in one complete loop. These measures were obtained in a $1.5\,\mathrm{m}$ width corridor, $0.9\,\mathrm{m}$ deep scenario in one direction and an open space to the other, with the initial orientation shown in Fig. 5. If we analyze one of the signals, for instance, the one corresponding to the central sensor (green line)[1], we observe 3 relative minima. The first and third minima correspond to the side walls (at approximately $800\,\mathrm{mm}$ and $700\,\mathrm{mm}$) and the second minimum to the ending wall (at approximately $900\mathrm{mm}$). The initial and final part of the graph corresponds with the measures where the robot is oriented towards the open part of the scenario, thus giving invalid measurements.

Using this information, we have developed the algorithm to build the occupancy map. The following algorithm has been designed taking into account the low resolution and accuracy of the sensors employed. First, for a single measure (one value for each of the three sensors for a given orientation) we pick the minimum reading of the three sensors, and consider this value as the distance of the closest object to the robot.

[1]Figures, code and multimedia material can be found at: http://agamenon.tsc.uah.es/Investigacion/gram/papers/LowCostRobotOrientation

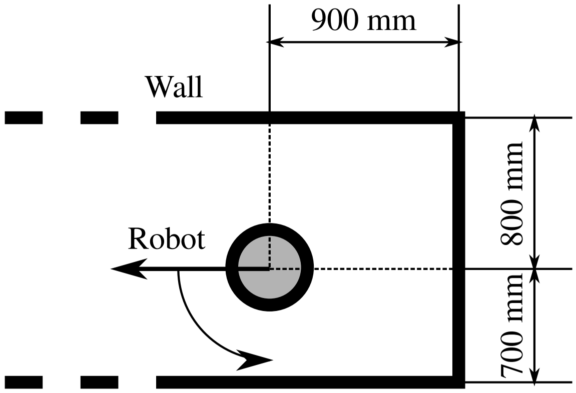


Fig. 5: A $1.5\,\mathrm{m}$ width, $0.9\,\mathrm{m}$ deep scenario. Initially the robot is facing the open side of the scenario, and turns counterclockwise to complete one loop.

If the second (third) minimum reading is within a given threshold from the first minimum, we consider the signal of both (the three) sensors coming from the same object. Using this procedure, we build the partial occupancy map seen from the rotation center, as can be seen in Fig. 6. This figure shows the partial occupancy map for a single measure, for different values returned by the sensors. For instance, in (a) the left, central and right sensors yield 500, 550 and 800 millimeters respectively. In this case, according with the algorithm, we assume the object is seen by both left and central sensor, so that the object is at a distance of $500\mathrm{mm}$ covering the angle overlapped by both sensors. If the three sensors give a distance greater than 1 meter, we assume there is no wall or object in front of the robot. Combining all the partial occupancy map constructed this way for all the orientations in one single loop of the robot gives the occupancy map seen from this point. In our case, we took 36 samples, i.e. 10 degrees between samples. For the scenario shown in Fig. 5, the occupancy map obtained is shown in Fig. 7. As we can see, only the closest parts of the walls are detected, leaving the corners open. This will be corrected with morphological operations, as described later. In our experiments MAP Scale=50 mm/pixel has been used.

Subsequent occupancy maps are obtained by straight translations of the robot to the open area, which, in our case, was 50 centimeters, and the global map by the combination of all those measurements is shown in Fig. 10. This image is 400x500 pixels that corresponds to an area of 20mx25m. The QR detected has been represented on the image, the map has irregularities that are not present in the scenario but it is enough for our propose.

IV. QR IDENTIFICATION AND ORIENTATION

QR stickers have been considered as the easiest way to configure a scenario, since no technical knowledge is needed for their installation by non-specialized staff. The QR sticker is used not only to identify destinations (QR minimum size is 50mm) but they also will be a reference.

Detection of QR codes are performed using the *zbar* library, which, besides the information codified inside the own code, also returns image coordinates of the four corners of the QR bounding square. Unfortunately, image coordinates returned by the *zbar* library are very inaccurate, which implies an error in robot localization in the range of tens of centimeters. In order to improve accuracy, we designed the following algorithm.

Fig. 8 (a) shows the original image of a QR code fixed on a wall captured with the RGB camera integrated in the robot. In Fig. 8 (b) the points corresponding to the corners of the QR bounding square returned by the *zbar* library are superimposed over the original image. As we can see, image coordinates are quite inaccurate, specially for the right-bottom corner. This inaccuracy is a typical imperfection of the QR detector algorithm itself as can be seen in Fig. 8 (c). It shows the lines joining two consecutive corners of the QR code.

To improve accuracy, we compute the edges (including orientation) of the image, as in fig. 8 (d). Using *zbar* image coordinates we generate four masks, each one created from a dilated version of the previous lines. For a specific line (say for instance, the horizontal top line in Fig. 8 (e)) we pick the pixels marked as edges whose orientation is within a given threshold from the line orientation. We repeat this procedure for the four masks. Fig. 8 (e) shows the set of pixels picked for the four lines, all in one single image.

The final step is the computation of the line parameters for the four set of points using standard methods from projective geometry. Fig. 8 (f) shows the lines computed using this algorithm. The cross point of two consecutive lines gives the image coordinates used for pose computation. Standard PnP algorithms were used for final camera pose estimation.

Since pose is computed with respect to QR coordinates system, to get actual robot pose, we need the world coordinates of the QR corners. This means basically that QR stickers must have known size and be placed in known positions. In our case, sticker size was fixed to $5\,\mathrm{cm}$, and placed in horizontal position 500mm above the floor, at one side of the door frame.

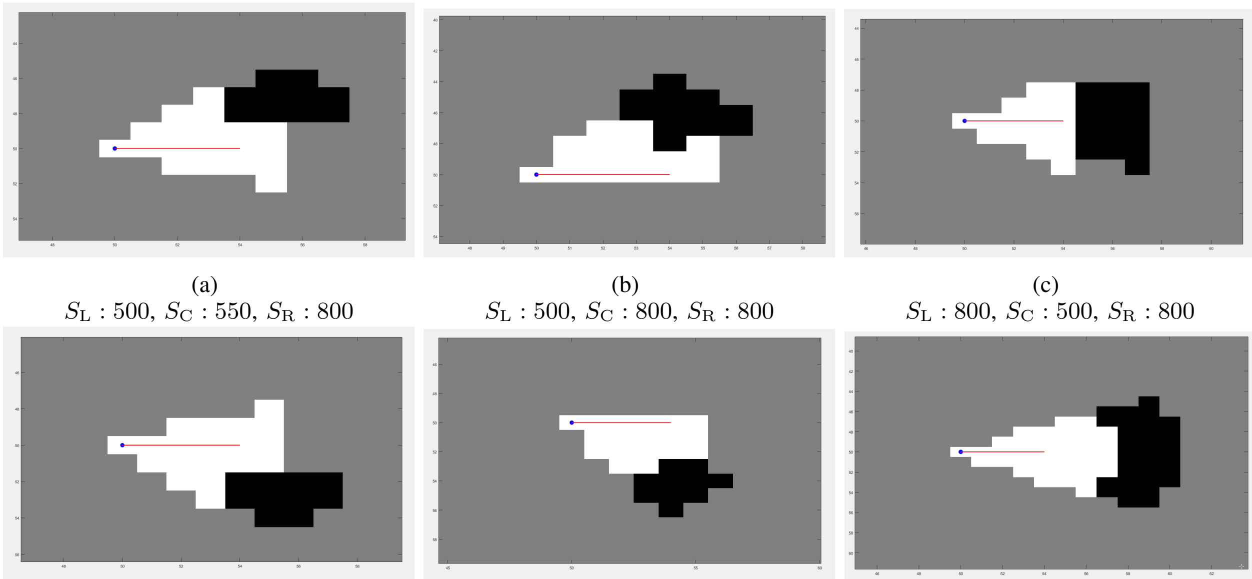
V. NAVIGATION

An occupancy map is used with a resolution of 50 mm/pixel (cell). The map pixel value can be 0 occupy cell, 255 for free space or 128 for unknown state. So the map is save as an image. QR positions represent possible destinations and they are saved during exploration in a text file with the following structure:

$$COD \quad XPOS \quad YPOS \quad XDIR \quad YDIR$$

COD is a code representing the given QR. (XPOS,YPOS) is the center position of the QR in the map and (XDIR,YDIR) is the end of a vector from the QR center indicating the orientation of the QR.

When a movement to a given destination is required the best route is calculated based on the reference map and the actual position of the robot. According to the route movement, commands are generated. Measurements using ultrasound sensors are taken while the movements are done. If measurements are



(a)
S_{L} : 500, S_{C} : 550, S_{R} : 800

(b)
S_{L} : 500, S_{C} : 800, S_{R} : 800

(c)
S_{L} : 800, S_{C} : 500, S_{R} : 800

(d)
S_{L} : 800, S_{C} : 550, S_{R} : 500

(e)
S_{L} : 800, S_{C} : 800, S_{R} : 500

(f)
S_{L} : 800, S_{C} : 800, S_{R} : 800

Fig. 6: Examples of construction of the occupancy map for a single measure. S_{L}, S_{C} and S_{R} are the measures, in mm, given by the left, central and right sensors respectively. Black pixels represent wall or objects, white pixels free space, and gray unexplored cells. (Straight lines represent robot orientation)

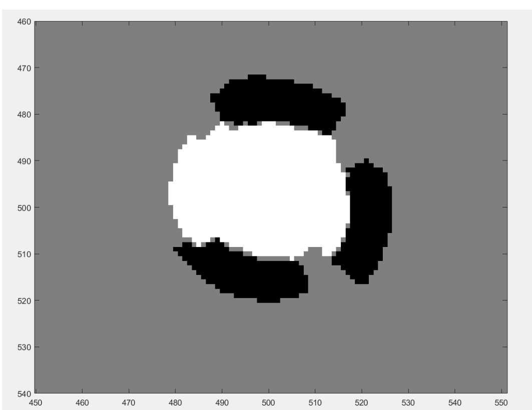


Fig. 7: occupancy map for one robot loop for the scenario shown in Fig. 5.

not compatible with the map, the reference map is updated and new route is obtained. QR search is perform during navigation and if a QR is detected, the robot position is updated based on QR reading. Periodically the route is calculated in order to prevent big deviations and if according to the encoder or QR readings a big deviation is detected, the position is corrected.

VI. Results

To test the proposed platform, we carried out the experiments in a $1.5\,\mathrm{m}$ width, $14.4\,\mathrm{m}$ length corridor and an open irregular area of aproximatly $32m^2$, where QR stickers has been fixed to some frame doors, to simulate possible destinations. Fig. 9 shows an image of this scenario, with some QR codes fixed in the door frames, for a detail image of QR check Fig. 1

An initial sweep is done to create the occupancy map for the whole scenario, and to localize the QR marks in it. Afterwards, the navigation gives the occupancy map and the QR positions. Finally Fig. 10 shows the final map, with ground truth QR position superimposed.

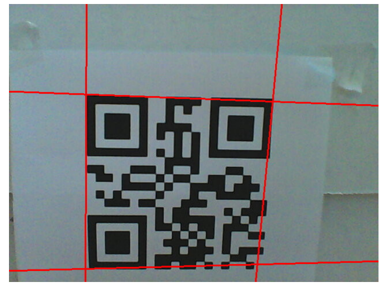
Several tests have been carried out, as a sumary we have ordered the robot to go to a destination approximately 4 meters apart from where it is and measure the deviation from where it should stop and the real stop position. Robot need to turn as the first movement because the initial position is looking at a reference QR of the first origin door. A voice message notice the origin room and it will inform the destination is reached with other voice message. After calibration, the same experiment has been repeated 10 times in order to get an idea of reproductibility of results. Those results are illustrated in table I, including (x,y) coordinates in the map coordinate system. Coordinates after turning are wirtten just one, because the maximum deviation from the given value was 1 mm. The final coordinates given by the robot and the real coordinates are presented together with the deviation.

VII. Conclusions and future work

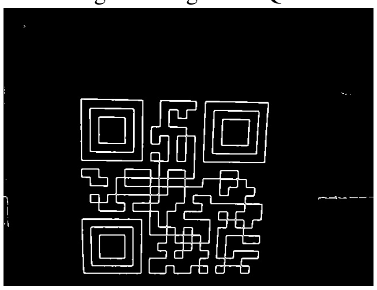
Our main goal was to obtain a mobile platform from scratch to orientate people with cognitive disorders inside a building. Movement, audio, exploration and navigation have been achieved with a low-cost platform.

(a)
Original image of a QR Code

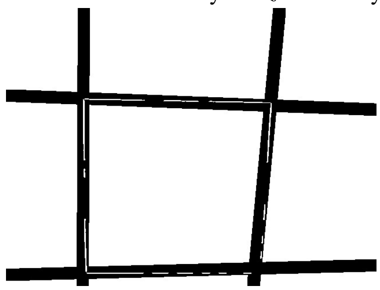
(b)
Points returned by the *zbar* library



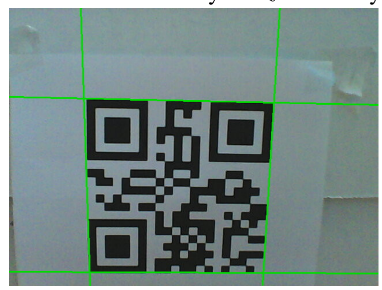
(c)
Lines returned by the *zbar* library



(d)
Edge detection of the QR Code



(e)
QR edge pixels



(f)
Lines computed using the algorithm

Fig. 8: Algorithm designed for the accurate computation of the camera pose with respect to the QR sticker.

	Initial x (mm)	Initial y (mm)				
	20400	13800				
	x after turn (mm)	y after turn (mm)				
	20303	13706				
Round	x Robot (mm)	y Robot (mm)	Real x (mm)	Real y (mm)	x Deviation (mm)	y Deviation (mm)
1	16282	13688	16271	13746	11	58
2	16296	13706	16281	13725	15	19
3	16308	13717	16295	13731	13	14
4	16309	13711	16297	13880	12	169
5	16324	13677	16359	13690	-35	13
6	16304	13732	16291	13594	13	-138
7	16323	13680	16299	13731	24	51
8	16311	13712	16302	13710	9	-2
9	16278	13682	16270	13856	8	174
10	16285	13748	16276	13679	9	-69
				Media:	7,9	28,9
				St. Deviation:	14,94	89,99

TABLE I: Results

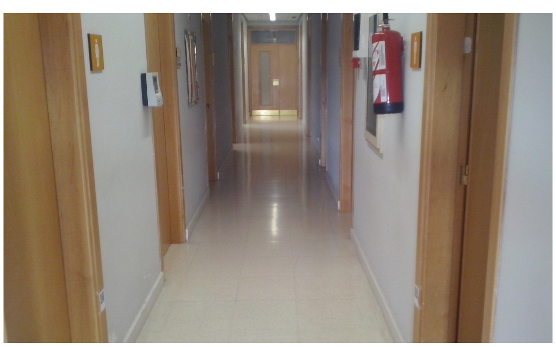


Fig. 9: Overview of the corridor where tests has been carried out. QR stickers can be noticed on the door frames (Check Fig. 1 for detail QR image.

Arduino board performs low-level funtion for movement and sensor control and Raspberry implements all other functionalities. A PCB board has been designed to include all the necessary electronics and it is used to connect Raspberry and Arduino. So the main conclusion is that the low cost platform can be navigated from one point to another in a given indoor scenario.

In future works, image processing funtion will be carried out in order to detect people and notify them that the robot is there or to move near the wall to let them pass. Visual odometry can be included in order to refine encoder odometry. Scene classification has been started in order to get a faster initialization of robot position after a switch on out of a reference point.

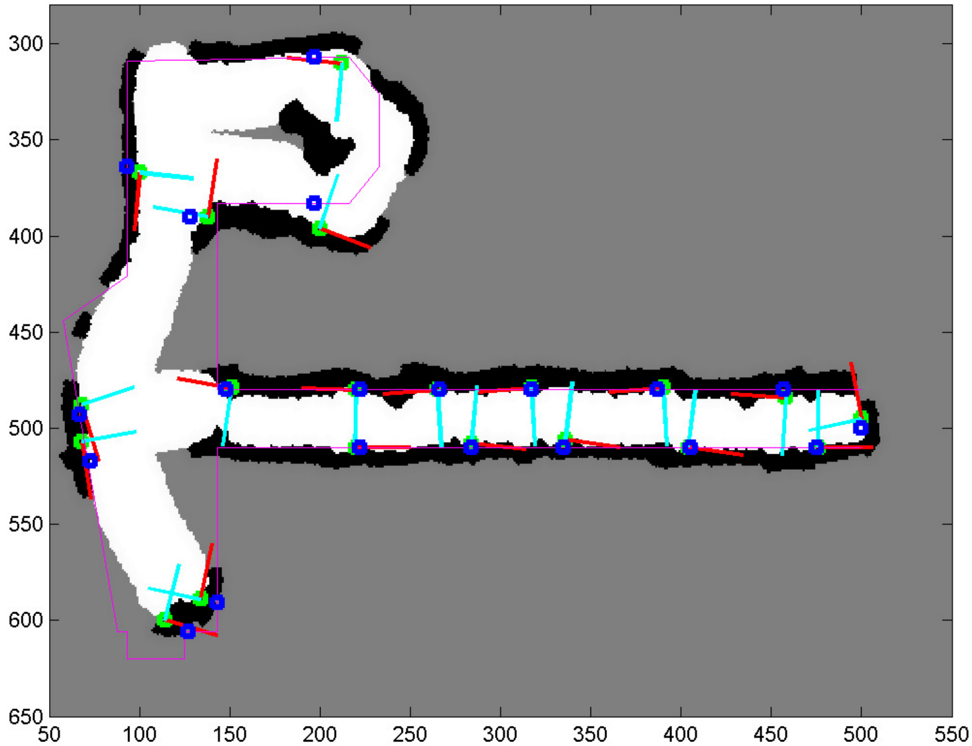


Fig. 10: General map. Black: walls or object; White: free space; Gray: unexplored cells; Green: detected QR landmarks; Blue: Ground truth QR landmarks; Purple: ground truth walls. Red and cyan lines represent the detected QR orientation.

Particle correction will be introduced in the next version in order to test improvements in mapping and navigation, although previous works are designed for more accurate sensors.

The final goal is to include different functionalities that can be used in a disable attention center to help assisting patients attending them.

ACKNOWLEDGMENT

This work was partially founded by the University Alcala research program through the project CCG2016/EXP-045 and CCG2016/EXP-042 and TEC2016-80326-R from the "Ministerio de Economía, Industria y Competitividad".

REFERENCES

[1] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Minerva: a second-generation museum tour-guide robot," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 3, 1999, pp. 1999–2005 vol.3.

[2] N. Hatao, R. Hanai, K. Yamazaki, and M. Inaba, "Real-time navigation for a personal mobility in an environment with pedestrians," in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, Sept 2009, pp. 619–626.

[3] C.-C. Lin and R. L. Tummala, "Mobile robot navigation using artificial landmarks," *Journal of Robotic Systems*, vol. 14, no. 2, pp. 93–106, 1997. [Online]. Available: http://dx.doi.org/10.1002/(SICI)1097-4563(199702)14:2¡93::AID-ROB4¿3.0.CO;2-O

[4] A. J. Briggs, D. Scharstein, D. Braziunas, C. Dima, and P. Wall, "Mobile robot navigation using self-similar landmarks," in *ICRA*, 2000.

[5] F. Boniardi, B. Behzadian, W. Burgard, and G. D. Tipaldi, "Robot navigation in hand-drawn sketched maps," in *Proc. of European Conference on Mobile Robots (ECMR)*, September 2015.

[6] K. Kamei, S. Nishio, N. Hagita, and M. Sato, "Cloud networked robotics," *IEEE Network*, vol. 26, no. 3, pp. 28–34, May 2012.

[7] S. S. Prabha, A. J. P. Antony, M. J. Meena, and S. R. Pandian, "Smart cloud robot using raspberry pi," in *2014 International Conference on Recent Trends in Information Technology*, April 2014, pp. 1–5.

[8] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb 2007.

[9] A. Doucet, N. de Freitas, K. P. Murphy, and S. J. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," *CoRR*, vol. abs/1301.3853, 2013. [Online]. Available: http://arxiv.org/abs/1301.3853

[10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Eighteenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002, pp. 593–598. [Online]. Available: http://dl.acm.org/citation.cfm?id=777092.777184

[11] J. Nordh and K. Berntorp, "Extending the occupancy grid concept for low-cost sensor-based slam," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 151–156, 2012.

[12] D.-L. Almanza-Ojeda, Y. Gomar-Vera, and M.-A. Ibarra-Manzano, "Occupancy map construction for indoor robot navigation," in *Robot Control*. InTech, 2016.