Ciencia e Ingeniería Neogranadina
ISSN: 0124-8170
ISSN: 1909-7735
Universidad Militar Nueva Granada

Muñoz Peña, Kevin; Bacca Cortés, Bladimir
GUI3DXBot: An Interactive Software Tool for a Tour-Guide Mobile Robot
Ciencia e Ingeniería Neogranadina, vol. 30, no. 1, 2020, pp. 59-74
Universidad Militar Nueva Granada

DOI: https://doi.org/10.18359/rcin.3644

Available in: https://www.redalyc.org/articulo.oa?id=91164537005

Ciencia e Ingeniería Neogranadina 2019 Vol. 30(1) enero - julio 2020 ■ ISSN: 0124-8170 · ISSN-e: 1909-7735 ■ pp. 59-74

Editorial Neogranadina

DOI: https://doi.org/10.18359/rcin.3644

GUI3DXBot: An Interactive Software Tool for a Tour-Guide Mobile Robot

Kevin Muñoz Peña[a] ■ Bladimir Bacca Cortés[b]

Abstract: Recently, mobile robots have begun to appear in public places. To perform these tasks properly, mobile robots should interact with humans. This paper presents the development of GUI3DXBot, a software tool for a tour-guide mobile robot. It focuses on the development of various software modules needed to guide users in an office building. In this context, GUI3DXBot is a client-server application, in which the server-side runs into the robot and the client-side runs into a 10-inch Android tablet. The GUI3DXBot server-side oversees the perception, localization-mapping, and path planning tasks. The GUI3DXBot client-side implements the human-robot interface that allows users to request/cancel a tour-guide service, show the robot a location in the map, interact with users, and teleoperating the robot in case of emergency. The contributions of this paper are twofold: a software module design to guide users in an office building is proposed and the whole robot system is well integrated and fully tested. GUI3DXBot is tested using software integration and field tests. Field tests are performed over a two-week period, and users are surveyed. Survey results show that users think GUI3DXBot is friendly and intuitive, goal selection was very easy, interactive messages are very easy to understand. 90 % of users found the robot icon on the map useful, users found path drawing on the map useful, 90 % of users found the local-global map view useful, and the guidance experience was very satisfactory (70 %) and satisfactory (30 %).

Keywords: Human-robot interaction; mobile robot; service robotics; tour-guide robot

Received: September 13th, 2019 **Accepted:** November 6th, 2019

Available online: July 15th, 2020

How to cite: K. Muñoz Peña and B. Bacca Cortes, "GUI3DXBot: An Interactive Software Tool for a Tour-Guide Mobile Robot", *Cien.Ing.Neogranadina*, vol. 30, no. 1, pp. 59-74, Nov. 2019.

a Universidad del Valle. E-mail: jhon.k.munoz@correounivalle.edu.co
ORCID: 0000-0002-4280-6358

b Universidad Nacional de Colombia, Colombia. email: bladimir.bacca@correounivalle.edu.co
Orcid: https://orcid.org/0000-0003-0113-4134

GUI3DXBot: Una herramienta de software *interactiva para un robot móvil guía*

Resumen: actualmente, los robots móviles comienzan a aparecer en lugares públicos. Para realizar estas tareas adecuadamente, los robots móviles deben interactuar con humanos. Este artículo presenta GUI3DXBot, un aplicativo para un robot móvil guía. Este artículo se enfoca en el desarrollo de los diferentes módulos de ***software*** necesarios para guiar a los usuarios en un edificio de oficinas. GUI3DXBot es una aplicación cliente-servidor, en la que el lado del servidor se ejecuta en el robot, y el lado del cliente se ejecuta en una tableta Android de 10 pulgadas. El lado del servidor de GUI3DXBot está a cargo de la percepción, localización-mapeo y planificación de rutas. El lado del cliente de GUI3DXBot implementa la interfaz humano-robot que permite a los usuarios solicitar o cancelar un servicio de guía, mostrar la localización del robot en el mapa, interactuar con los usuarios, y teleoperar el robot en caso de emergencia. Las contribuciones de este artículo son dos: se propone un diseño de módulos de ***software*** para guiar a los usuarios en un edificio de oficinas, y todo el sistema robótico está bien integrado y completamente probado. GUI3DXBot fue validada usando pruebas de integración y de campo. Las pruebas de campo fueron realizadas en un periodo de dos semanas, y una encuesta a los usuarios fue llevada a cabo. Los resultados de la encuesta mostraron que los usuarios piensan que GUI3DXBot es amigable e intuitiva, la selección de metas fue fácil, pudieron entender los mensajes de interacción, 90 % de los usuarios encontraron útil el ícono del robot sobre el mapa, encontraron útil dibujar la ruta planeada en el mapa, 90 % de los usuarios encontraron útil la vista local-global del mapa, y la experiencia de guía fue muy satisfactoria (70 %) y satisfactoria (30 %).

Palabras clave: Robot guía; robot móvil; interacción humano-robot, robótica de servicio

Introduction

People guidance is a common service that mobile robots have begun to provide. It is rendered in public places such as hospitals, hotels, exhibitions, supermarkets, office buildings, and museums [1]. For example, a mobile robot can be used to guide patients to examination rooms or doctor's offices in a hospital; to take customers to their rooms and carry their luggage in a hotel; to find products for the newest customers in a supermarket, and to reduce the risk of missing connections in an airport [2]. Using mobile robots to perform people guidance tasks has the following advantages: on the one hand, employees can focus on their jobs without interruption and fulfill more important responsibilities, as people guidance and information providing could be repetitive activities; on the other hand, mobile robots can carry more weight without health risks and be used as telepresence platforms and shared surveillance systems while performing people guidance tasks.

Many components are important in performing a people guidance task, for instance, the ability to self-localize and navigate in a dynamic environment; i.e., the mobile robot must use its own sensors to estimate its position and guarantee safe guidance to the user's goal. Also, a human-robot interface should provide enough information to people and collect feedback data from users, while an emergency system should be aware of situations such as unreachable goals, battery level and robot position lost.

This paper focuses on the development of GUI3DXBot [3] software tool and all the software modules needed to guide users in an office building, specifically the School of Electrical and Electronic Engineering at Universidad del Valle so that the newest users can find professor's offices, electronics labs, and research labs. To do so, GUI3DXBot was conceived as a client-server application. The server side oversees the perception, localization, mapping, and path planning tasks. These tasks use the LMS200 laser range finder (LRF) as the main sensor and were implemented in the ROS framework [4]. The GUI3DXBot client-side implements the human-robot interface to offer a friendly and interactive user interface. The GUI3DXBot user interface allows users to request/cancel a tour-guide service, show the current robot location in the environment map, interact with users using a robot avatar, communicate with the GUI3DXBot server-side, and teleoperate the mobile robot in case of emergency.

This paper is structured as follows: Section 1 presents related works, Section 1.1 details the experimental platform, Section 2 described the GUI3DXBot server-side and Section 3 the client-side, Section 4 shows the results obtained, and Section 5 contains the conclusions.

Related works

Nowadays, there are many academic and commercial implementations of tour-guide solutions. Table 1 shows a review of previous works related to

Table 1. State-of-the-art summary of tour-guide solutions

Ref.	Sensor	Map	Loc. Method	Obst. Av.	HRI	Env.	Application
[1]	Camera, LRF	Sub-mapping	Sub-map – SLAM	Pedestrian based	Touchscreen	Indoors	Reception, guidance
[2]	RGBD, LRF	Occupancy grid	SLAM, CTMap	---	Touchscreen, barcode reader	Indoors	Airport guidance
[5]	Camera, LRF	Occupancy grid	Particle Filter	μDWA	Touchscreen	Indoors	Museum tour guide
[6]	Camera	Topological	Bayes	Reactive algorithm	Pushbuttons	Indoors	Museum tour guide
[7]	RGBD, LRF	Occupancy grid	Particle Filter	DWA	Touchscreen	Indoors	Office tour guide

Continued

Ref.	Sensor	Map	Loc. Method	Obst. Av.	HRI	Env.	Application
[8]	Smartphone camera, Ultrasonic	Topological	QR recognition	Reactive algorithm	Mobile app, touchscreen	Indoors	Exhibition tour guide
[9]	Camera, LRF	Sub-mapping	Sub-map - SLAM	---	Voice commands	Indoors	Campus tour guide
[10]	LRF	Metric map	SLAM-EKF	---	Touchscreen, voice comm., web	Indoors	Exhibition tour guide
[11]	LRF, GPS	Occupancy grid	Particle filter	Reactive algorithm	Touchscreen	Outdoors	Pedestrian assistant
[12]	Sonar, RFID	Topological	RFID recognition	Reactive algorithm	Touchscreen	Indoors	Exhibition tour guide
[13]	RFID, LRF	Metric map	EKF	Fuzzy logic	Touchscreen, mic	Indoors	Exhibition tour guide
[14]	Camera, LRF	Occupancy grid	Particle filter	Potential fields	Voice commands	Indoors	Museum tour guide
[15]	LRF, omni- camera	Occupancy grid	Particle filter	VFH	Touchscreen, eye tracking	Indoors	Shopping guide
[16]	Camera, LRF	Occupancy quadtree	Particle filter	DWA	Voice commands, mobile app	Indoors	Shopping guide
[17]	RFID, camera, LRF	Occupancy grid	Particle filter	DWA	Touchscreen	Indoors	Campus tour guide
[18]	RFID, LRF	Metric map	LSM-RFID detection	---	---	Indoors	Exhibition tour guide

Source: The authors.

this paper, comparing sensors, type of map used, localization method, obstacle avoidance method, human-robot interface (HRI), used, deployment environment, and main application.

According to Table 1, the main sensor used is the laser range finder (LRF) with different types of cameras such as monocular, RGBD and omnidirectional. However, special localization sensors are also employed such as RFID [12], [13], [17], [18] and QR codes [8]. These kinds of sensors are used to obtain an initial localization hypothesis which is refined by LRF or vision sensors.

Another important part of these tour-guide solutions is the map used, since depending on it, the tour-guide robot can achieve proper environmental representation. This map can be updated over long-term usage. Then, Table 1 shows that, in decreasing order of importance, the occupancy grid, topological, sub-mapping and metric maps are preferred for tour-guide solution implementations.

Map building and localization methods are crucial to implementing tour-guide solutions. Table 1 shows that particle filter-based methods are the SLAM (Simultaneous Localization and Mapping) approach which better suits this kind of application. This is because particle filters perform better in populated environments, which are prone to localization errors due to occlusions and substantial changes in the illumination of the environment. However, in [2], [7], [14] and [16] the SLAM approach is modified in order to include social awareness for robot navigation. This property is important to compute robot motion and capture the correct motion patterns of pedestrians. Long-term navigation in real environments is another important challenge of tour-guide robots. This aspect is considered in [5], [9], [11] and [15], in which the SLAM approach and the planning method were implemented considering a full-time job with minimal human intervention.

Sensors and technical issues related to SLAM are important, but if all the information generated by these methods cannot be effectively communicated to humans, the tour-guide service cannot be accomplished. For this reason, HRI is

a very important part of any tour-guide robotic solution. Table 1 shows how HRI has evolved over many years, starting with simple push buttons [6] and then using virtual maps [10]; going through touchscreen interfaces, as reported in most works in Table 1, and the capability of understanding simple and complex voice commands, as proposed in [9], [10], [14] and [16]; and finally, including special sensors which depend on the application as in [2], as well as detecting pedestrians intentions using eye-tracking [15].

In this paper, the GUI3DXBot software tool is presented. GUI3DXBot controls a tour-guide robot with the aim of guiding the newest visitors around the School of Electrical and Electronic Engineering at the Universidad del Valle. Then, comparing the works presented in Table 1 and GUI3DXBot, it is important to highlight the following remarks:

- GUI3DXBot uses an LRF as the main sensor to build the map of the environment and localize the mobile robot.
- GUI3DXBot uses Gmapping [19] in order to build an occupancy grid map with a particle filter as a localization method.
- GUI3DXBot uses the trajectory rollout algorithm for obstacle avoidance.
- GUI3DXBot uses a touchscreen as a human-robot interface. Through this friendly user interface, users about emergency situations (unreachable goals, low battery level, or robot position lost) and select goals, showing users the path to their goals and its real-time evolution on a map of the environment.
- GUI3DXBot works in indoor environments like an office tour-guide robot application.

Therefore, this paper makes two main contributions: first, it proposes a software module design to guide users in an office building, since most works presented in Table 1 (except [2], [9] and [11], and [15]) have custom software designs without considering standard middleware such as ROS [4]. And second, the whole robot system is well integrated and fully tested.

System configuration

Fig. 1 shows the hardware components needed to run the GUI3DXBot software tool. On the right side, there is the Pioneer 3DX robot base, the LMS200 LRF as the main mapping and obstacle avoidance sensor, the Wi-Fi router as the remote communication interface, and the 10-in Android device which displays the user interface depicted on the left. ActivMedia robotics built the Pioneer 3DX robot base that has the following technical specifications: a differential drive mobile robot, two 500 ticks/rev incremental encoders, eight frontal sonar sensors, one on-board computer running Ubuntu 14.04 and ROS Hydro, a three 12V/7Ah battery pack, and a metal support to hold sensors on top.

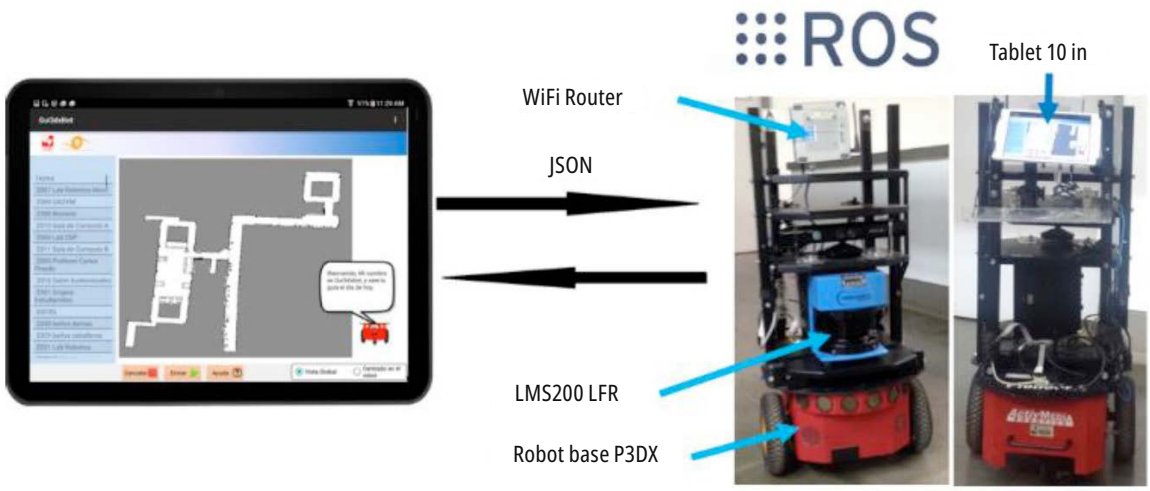


Fig. 1. System configuration for the GUI3DXBot software tool.
Source: The authors.

The communication protocol between the mobile robot and the Android device is performed using JSON objects through web-sockets and using the Wi-Fi router as a communication interface.

GUI3DXBot server-side software development

Mobile robot localization and mapping are essential software modules used by service robots. To do so, in the context of office tour-guide robots, this section describes the development of the GUI3DXBot server-side software which includes global localization, communication with the client-side, path planning, and local and global navigation tasks.

Design requirements

GUI3DXBot was developed using the RUP methodology [20]. Only functional and non-functional requirements and class diagrams are presented due to space limitations. The GUI3DXBot server-side functional and non-functional requirements are described as follows:

- *Functional requirements*: The GUI3DXBot server side should be able of building a map of the environment using range sensors, localizing itself on the map, planning a path to the goal requested by users, performing local planning in case of obstacles, navigating to its home location at the end of the office tour-guide service, and raising alerts for users in case of emergency situations.
- *Non-functional requirements*: The GUI3DXBot server-side works on Ubuntu 12.04 and was developed on ROS Hydro, the robot base is a Pioneer 3DX, and the range sensor is the SICK LMS200.

Mobile robot mapping, localization, local and global navigation

The functional requirements described above were implemented using a set of software modules that run on the GUI3DXBot server side. Then, the GUI3DXBot server-side is in charge of localization, local and global navigation, and communication with the HRI.

To start offering guidance services using a mobile platform, a previous mandatory step is needed: mapping the environment. To do so, Fig. 2a shows the computational graph which represents the mapping process. This figure, as well as Fig. 2b, are common ways to represent the algorithm implemented when ROS is used. Then, the mapping process was performed using the *GMapping* package, which requires odometry information provided by the mobile robot through the *RosAria* package, range data provided by the laser sensor, and transformation between the laser sensor and the mobile robot platform. The mapping process is performed once unless the environment structure changes. Then, this map is further used for the people guidance service.

The people guidance service which implements the GUI3DXBot software tool is constantly localizing the robot in the map, planning the path to the goal requested by users, and avoiding obstacles. Fig. 2b shows the computational graph that relates the software modules in charge of these tasks with each other. As observed in Fig. 2b, the *move_base* node is an important part of the robot navigation software; it requires odometry readings provided by the *RosAria* package, the map of the environment, current sensor readings, and the current robot position. The *move_base* node oversees robot navigation, which includes local and global path planning; however, these features are useless without a current robot position.

Then, the first thing the GUI3DXBot software tool should do is to localize the mobile platform. To do so, the AMCL package is used; it implements the adaptive Monte Carlo Localization approach [4], which requires robot odometry readings provided by the *RosAria* package, laser scan readings provided by the *SICKLms* package, and relative transformation between the laser sensor and the mobile base to localize the mobile robot.

Secondly, the GUI3DXBot software tool should plan the robot path from the current position to the goal requested by users. To do this, the package *Navfn* [4] was used as part of the node *move_base*. Once the user's goal is known, this information and the current robot position are introduced in the *Navfn* package to compute the global path.

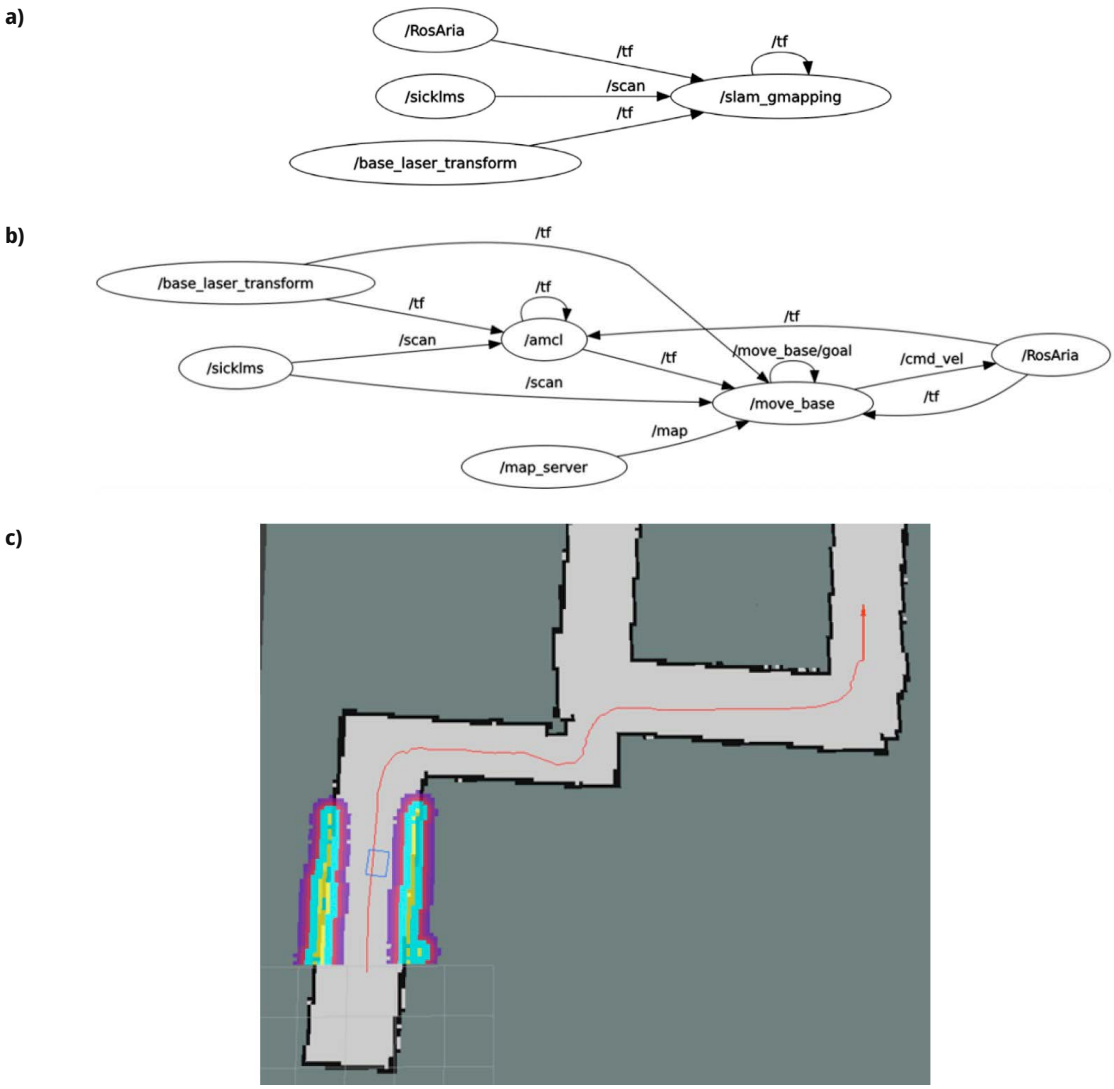


Fig. 2. a) Computational graph of the mapping process. b) Computational graph of the localization and navigation process. c) Cost map for local and global path planning.

Source: The authors.

The global path is computed by the *Navfn* package using Dijkstra's algorithm. Fig. 2c shows an example of this global path.

The people guidance service is commonly deployed in public spaces, and obstacle avoidance is essential to guarantee a safe route to the user's goal. Then, global path planning could be modified locally when the mobile robot faces obstacles. To implement this feature, the *move_base* node has another important package called *costmap_2d* which implements the trajectory rollout algorithm [21] in order to compute the local robot trajectory considering obstacles in the environment. Fig. 2c shows an example of the local cost map. Using this algorithm, different trajectories are evaluated at each robot position based on obstacle distribution and the robot kinematic model. Each trajectory is evaluated using a cost function which includes sensor readings, local static maps, the relative position of inflated obstacles with respect to the robot, robot footprint, size of the local map, among others. In the end, the low-cost trajectory is selected.

As observed in Fig 2a and 2b, graph nodes define a server-side software module design that

can be replicated in most platforms compatible with ROS. In this work, this design is used to provide the people guidance service in an office building, but it can be adapted to many other applications. In addition, the entire server-side software module was implemented using new and existent ROS functionalities. New ROS functionalities include transform nodes, odometry sensor handler, robot base handler, and all sensor topics (laser and point cloud). Finally, the server-side software module has no GUI, so all parameters are configured either by command line or using the GUI3DXBot user interface.

GUI3DXBot user interface

Section 2 described the GUI3DXBot server-side, which handles information that mobile robots can understand. This section shows the software modules of the GUI3DXBot user interface, whose purpose is to send user requests and receive feedback and status information from the server-side.

Design requirementS

The GUI3DXBot user interface was also developed using the RUP methodology [20]. Only functional and non-functional requirements and class diagrams are presented due to space limitations. The GUI3DXBot client-side functional and non-functional requirements are described as follows:

- *Functional requirements*: The GUI3DXBot user interface should display the current map and all available goals to users, receive the user's goal, trace the path to the user's goal on the map while the guidance service is running, provide current robot position on the map while the guidance service is running, show the local or global view of the environment, receive the user's cancel order if any, deliver status information using a robot avatar, and give emergency information using a robot avatar.
- *Non-functional requirements*: The mobile device minimum OS version is 5.0.2, the server-side was implemented on ROS Hydro, and the mobile robot platform was a Pioneer 3DX.

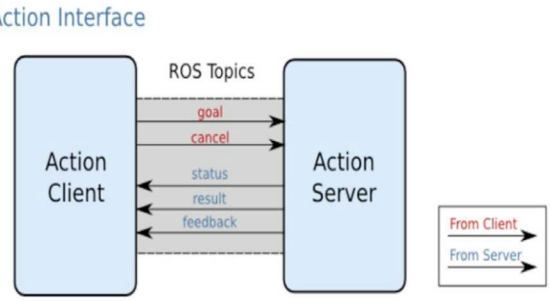
GUI3DXbot user interface development

The functional requirements described above were implemented as GUI3DXBot HRI software modules. This HRI exchanges messages between the GUI3DXBot server-side (action server) and the GUI3DXBot user interface (action client), as shown in Fig. 3a. As described in Section 2, the GUI3DXBot server-side needs to know the requested user's goal or start/cancelation order to initiate the planning and navigation process. In turn, the HRI needs to obtain status information such as current robot position, current local/global map, or the list of different goals available; in addition, the HRI needs to know feedback information such as messages to be displayed by the robot avatar (start/cancellation/successful end of a guidance mission) and emergency alerts (low battery, unreachable goals).

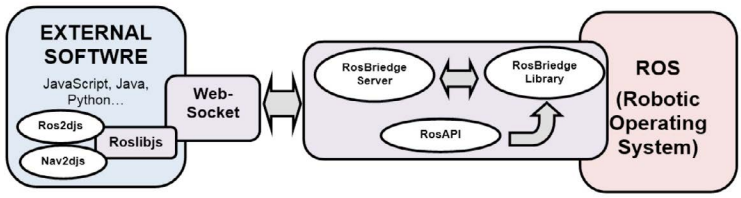
All this information is exchanged using JSON objects, which are sent using a web-socket connection between the GUI3DXBot server and client sides. This is depicted in Fig. 3b, which also shows the server-side technologies involved in this communication, namely, *RosBridge Library*, which is responsible for taking JSON objects and sending them to ROS, and vice-versa; *RosBridge Server,* which implements the web-socket connection using the *RosBridge* protocol [22]; and *RosBridge API,* which is used by clients to access reserved services of ROS libraries. On the client-side, *Roslibjs* implements the *RosBridge* protocol to provide the ROS functionalities for this work such as topic publication/subscription, calling ROS services, action servers, performing the transformation, etc. Internally, the HRI uses *Ros2djs* and *Nav2djs* to display the environment map, robot position, robot path, and robot goals. These functionalities were implemented in the GUI3DXBot user interface.

Fig. 3c shows the class diagram of the software module implemented by the GUI3DXBot user interface made up of two Android activities, namely, the navigation or main activity and the teleoperation activity. The *NavScreen* class implements all the functional requirements described in Section

a)



b)



c)

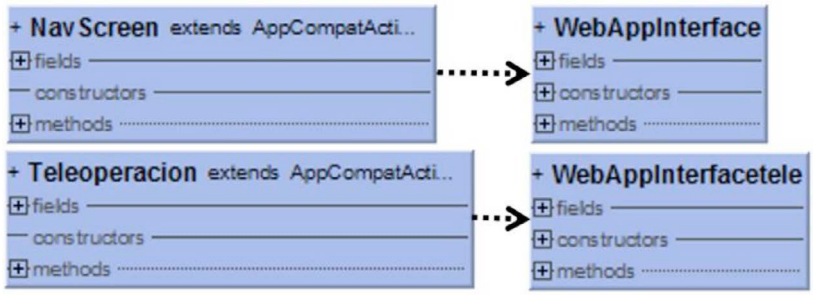


Fig. 3. a) Communication structure of the action interface between the *move_base* node and the GUI3DXBot user interface [4]. b) Conceptual diagram of the communication structure software. c) Class diagram of the GUI3DXBot user interface.

Source: The authors.

3.1, while the *Teleoperation* class implements the robot teleoperation functionality, which is used when the mobile robot requires human assistance. In this case, the GUI3DXBot user interface requests a password to move the mobile robot manually. Both activities use the *WebAppInterface*, which acts as a bridge between the GUI3DXBot user interface, JavaScript, and Android. In this way, *WebSocket* communication between the GUI3DXBot user interface and the server-side can be established.

As mentioned, the GUI3DXBot user interface has two activities navigation (*NavScreen* class) (Fig. 4) and teleoperation (Fig. 5). The navigation activity is divided into four parts, namely, goal list (number 1 in Fig. 4), environment map (number 2 in Fig. 4), robot avatar (number 3 in Fig. 4), and controllers bar (number 4 in Fig. 4). The four parts of this activity contain a total of eight graphical components and their corresponding event handlers which were developed to implement the GUI3DXBot user interface.

The scrollable goal list contains all the places to which the mobile robot can guide people. Here, the user just selects the goal location and the guiding process starts. The GUI3DXBot server-side continuously sends the environment map and shows mobile robot trajectory, current robot location, and goal location.

The robot avatar interacts with users and, through it, GUI3DXBot displays and sends messages using an Android sound synthesizer. These messages include asking users if they want the guidance service, raising emergency alerts (low battery, unreachable goals), and notifying users they have successfully reached the goal requested.

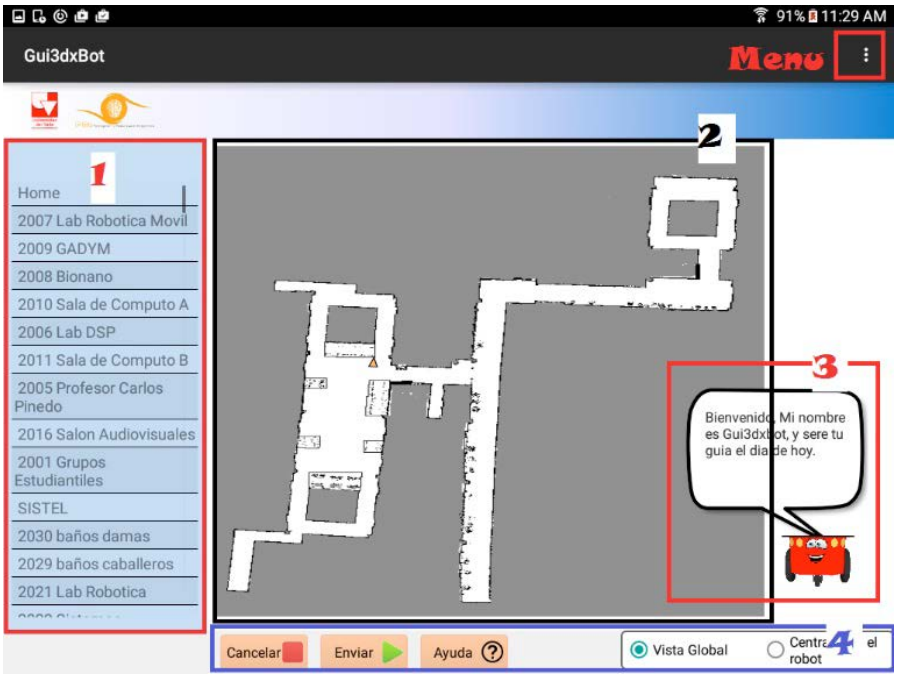


Fig. 4. Navigation activity (NavScreen) of GUI3DXBot.

Source: The authors.

Using the controllers bar, users can start a guidance service ("Enviar" button), cancel a guidance service at any moment ("Cancelar" button), read how the guidance service works ("Ayuda" button), and change the environment map view to global or local at any time during the tour.

Fig. 5 shows the teleoperation activity that is activated using the Menu in the upper right of the

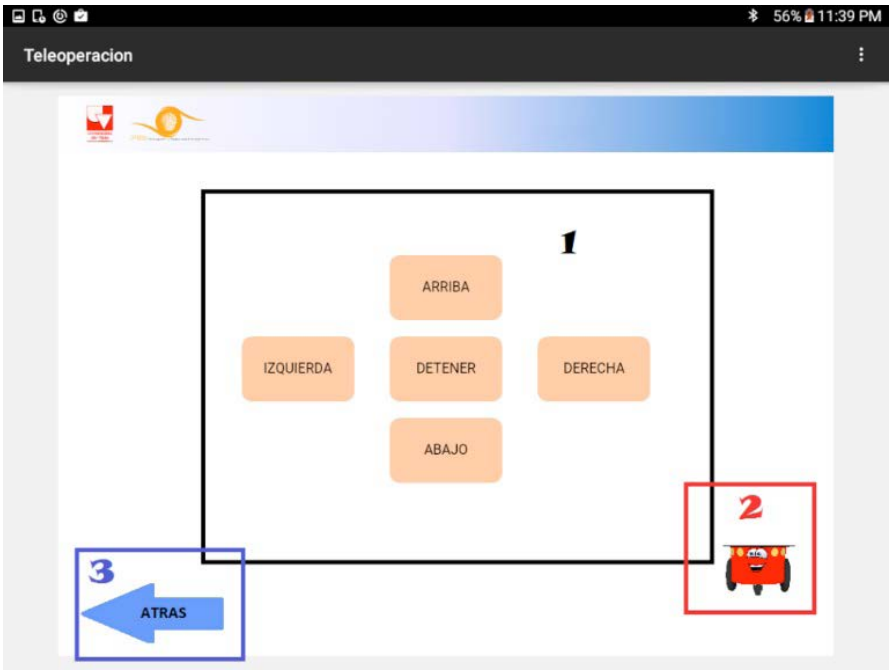


Fig. 5. Teleoperation activity of GUI3DXBot.

Source: The authors.

GUI3DXBot GUI. The Menu also allows for updating the goal list when the environment has changed. The teleoperation activity is accessible once the user password is validated. This Android activity is composed of three parts, namely, teleoperation controls (number 1 in Fig. 5), robot avatar (number 2 in Fig. 5), and back button (number 3 in Fig. 5). The three parts of this activity contain a total of seven graphical components and their corresponding event handlers which were developed to implement the GUI3DXBot user interface.

Teleoperation controls allow users to command the motions of the mobile robot. The robot avatar displays and raises sound alerts while the user is commanding the robot motion. The back button returns to navigation. It is worth noting that teleoperation is executed in case of an emergency, when the mobile robot needs human assistance.

Results and discussion

The functionality of GUI3DXBot and its software modules was verified using the following tests: integration tests using the RUP methodology to validate functional requirements, and real-world trials over a period of two weeks in the School of Electrical and Electronic Engineering at the Universidad del Valle. After the guidance service was offered, the user was asked to answer a survey about their experience with GUI3DXBot. This section describes the experimental setup and the results obtained from real-world trials under which the robot system proved to be well integrated. Find an example of the guidance service offered at https://www.youtube.com/watch?v=bUg2RNfpRDQ

Experimentation setup

At the very beginning, the mobile robot has no map of the environment. Then, to offer the guidance service, this map should be computed in advance. This section describes the procedure for such purpose as follows:

1. The mobile robot platform depicted in Fig. 1 is powered on, and the position where it starts is taken as the home location.
2. Using a command terminal, the *RosAria* and *sicklms* nodes are launched to obtain laser data and command the robot motion.
3. Using a command terminal, the *teleop_nav* node is launched to get user keyboard inputs, move the robot around the environment, and save all the data gathered in a *bagfile*.
4. Once data collection finishes, the software module shown in Fig. 2a is executed to compute the environment map. This map is an occupancy grid map with a resolution of 0.08 m.
5. Using an image processor, the PNG file that contains the environment map is opened and refined, i.e., adding forbidden zones, clearing known zones, etc. 6. Finally, using the *rviz* home position and goal coordinates are configured.

Using this procedure, the map corresponding to the second floor of buildings 353 and 354 of the School of Electrical and Electronic Engineering at the Universidad del Valle was computed, as depicted in Fig. 6.

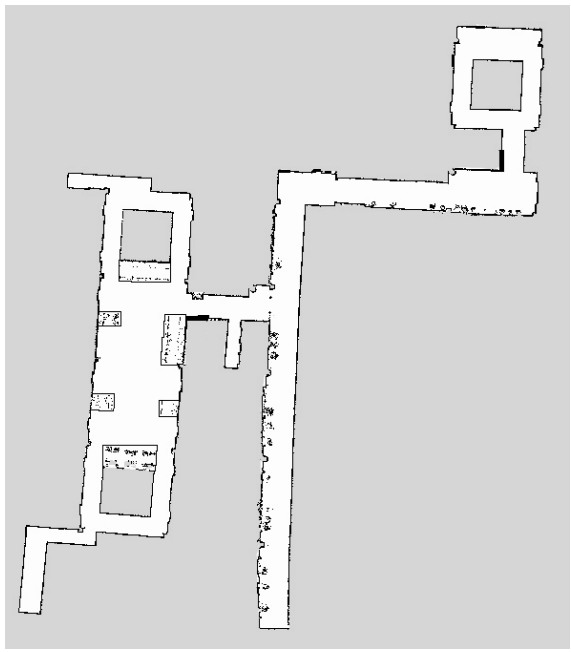


Fig. 6. Map of the second floor of buildings 353 and 354 of the School of Electrical and Electronic Engineering at Universidad del Valle.

Source: The authors.

Tests and results

The procedure described in Section 4.1 is performed only once, but, if the environment changes, this procedure needs to be repeated. The real-world trials reported in this section were carried out over two weeks. The environment, where the guidance service was offered, was not altered, that is, people walked normally over these places and did their normal activities. Trials were performed during business days to guarantee real conditions of experimentation and that the robotic system is well integrated and fully tested.

Over these three weeks, 33 interactions with humans were registered. At the end of each human interaction, the user was asked to answer a survey. This survey aimed at evaluating the experience with the guidance service. The questions asked, as well as their answers and quantitative values, are described as follows:

1. In general, what do you think about the GUI3DXBot GUI? **Answers**: Very nice (5), Nice (4), Slightly nice (3), Unpleasant (2).
2. How intuitive is the GUI3DXBot GUI interaction? **Answers**: Very intuitive (5), Intuitive (4), Slightly intuitive (3), Counterintuitive (2).
3. How good is the information provided by the GUI3DXBot GUI about your goal? **Answers**: Excellent (5), Average (4), Below average (3), Very poor (2).
4. How easy was it to choose a goal from the list in the GUI3DXBot GUI? **Answers**: Very easy (5), Easy (4), Difficult (3), Very difficult (2).
5. How good is the information provided by the robot avatar to use the GUI3DXBot application? **Answers**: Excellent (5), Average (4), Below average (3), Very poor (2).
6. How easy is it to understand the messages sent by the robot avatar? **Answers**: Very easy (5), Easy (4), Difficult (3), Very difficult (2).
7. How useful is the robot icon on the map? **Answers**: Very useful (5), Useful (4), Slightly useful (3), Useless (2).
8. How useful is the robot trajectory depicted on the map? **Answers**: Very useful (5), Useful (4), Slightly useful (3), Useless (2).
9. How useful is the global and local map view? **Answers**: Very useful (5), Useful (4), Slightly useful (3), Useless (2).
10. How often did the guidance service have difficulties? **Answers**: Often (2), Sometimes (3), Rarely (4), Never (5).
11. In general, how do you rate the experience with the guidance service provided? **Answers**: Very satisfactory (5), Satisfactory (4), Slightly satisfactory (3), Unsatisfactory (2).
12. What would you change or add to the GUI3DXBot application?

Respondents were distributed as follows: 33 % were electronic engineering students, 12 % were Master of Science students, and 55 % were non-engineering students (basic sciences, industrial design, and medicine students). It can be observed that most users were non-engineering students, which is important in considering different perceptions of GUI3DXBot.

Fig. 7 shows the results obtained from users' answers, which were analyzed based on seven aspects, namely, GUI, goals, human-robot interaction using the robot avatar, computed trajectory, map visualization, problems with the guidance service, and user comments. Using Questions 1 and 2 it is possible to know how friendly the GUI3DXBot GUI is. Results of Question 1 show that users like the appearance and structure of the GUI3DXBot GUI. But, to understand the results of Question 2, it is worth saying that 52 % of users who rated the GUI as *Intuitive* got confused with the "Cancel" button; however, once they experimented with the GUI3DXbot, they resolved their doubts quickly.

Questions 3 and 4 provide evidence of how easy it is to choose a goal in the GUI3DXBot software tool. These results are satisfactory and this is because the goal list is compact, goals are always present in the GUI so users can cancel a guidance service and select another goal, and goal descriptions are meaningful. Human-robot interaction is important in a guidance service, then Questions 5 and 6 reveal if users could understand the GUI3DXBot feedback messages. As observed in Fig. 7, users liked the robot avatar and could understand its messages.

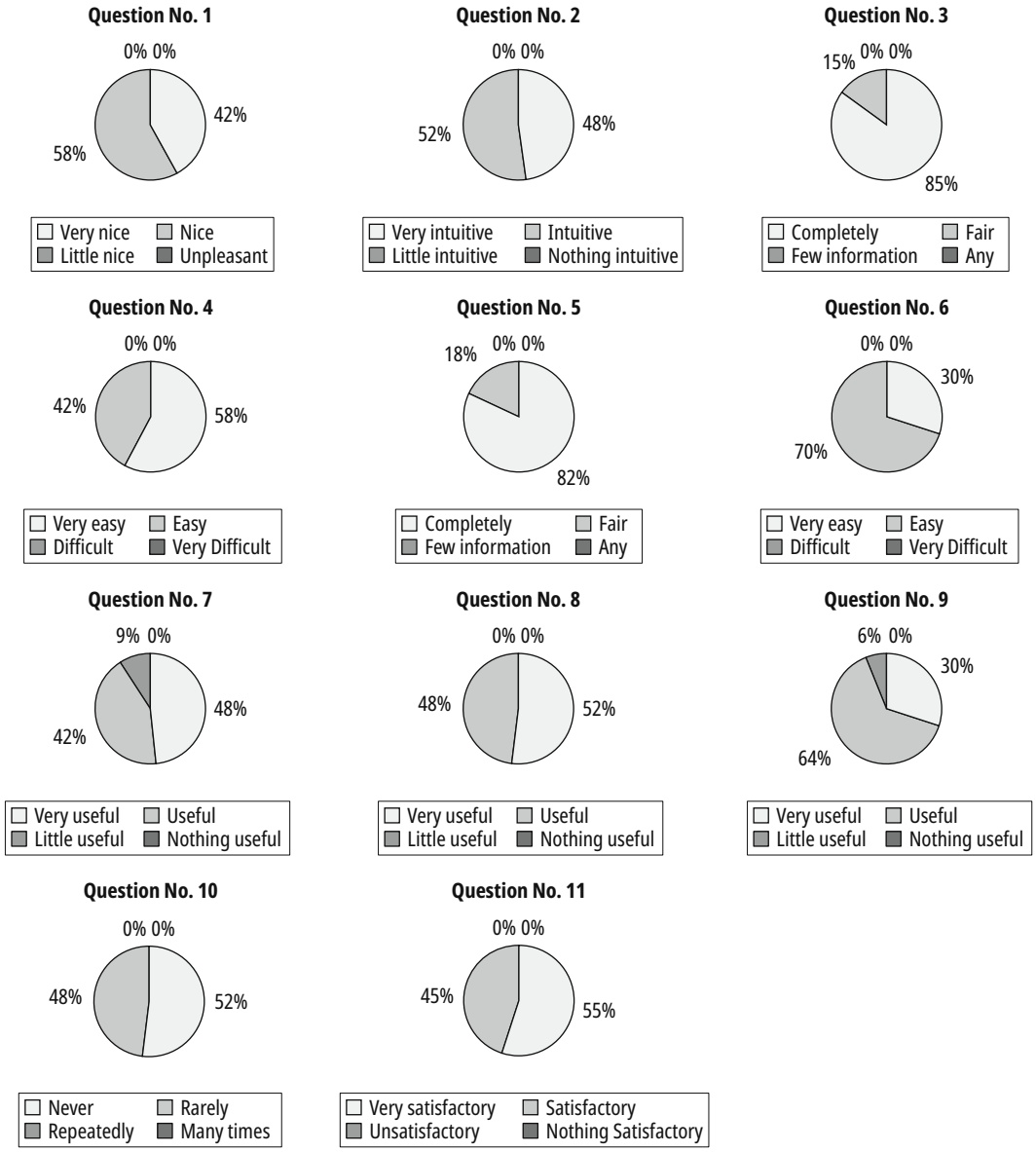


Fig. 7. Survey results.

Source: The authors.

Another type of human-robot interaction tested is the proper status information. For this, Question 7 helps to understand if the robot icon on the map was meaningful. Fig. 7 shows that 90 % of respondents found the robot icon useful, which could be located other users in the environment. However, 10 % found the robot icon slightly useful, as they argued this icon should be more attractive to people. Meanwhile, Question 8 shows evidence of how important it is to show what actions are planned by the guide robot. In this sense, Fig. 7 shows satisfactory results and users said that plotting the robot trajectory helps them to memorize the path for some other time.

Guidance services do not only allow users to reach their goals but also encourage them to orient themselves. Question 9 evaluates if users liked the local and global view of the environment map. Fig. 7 suggests that users used the functionality of changing from local to a global view of the

environment for the guidance service; however, 6 % argued it is slightly useful since they preferred the local view instead of the global view.

Evaluating the user experience of interacting with service robots is important in observing the acceptance of this kind of application. Questions 10 and 11 allow perceiving this aspect. The fewer failures the more acceptance in the guidance service. In this case, the survey reports 52 % of users had no failures in the service. Since the environment was not altered and trials were performed in business days, corridors and halls were full of people passing by. Then, failures perceived by users were robot motion stop and start-stop motions due to trajectory re-planning, but in the end, the guide robot reached its goal. These are exceptional cases, but in general, users reported a very satisfactory (55 %) and satisfactory (45 %) guidance experience as reflected in the results of Question 11.

Finally, additional comments made by users were: "Increasing the robot motion speed," "Increasing the size of buttons for better access while the robot is moving," "Adding voice commands," "Improving the robot icon on the map, as it is not friendly nor visible," and "When the mobile robot is moving, it is difficult to access the GUI buttons." Most of these issues can be solved using a bigger touchscreen. Motion speed depends on the CPU power since it must compute local cost maps for obstacle avoidance; therefore, a more powerful CPU could resolve this issue. Moreover, changing the human-robot interaction using voice commands is an interesting improvement for further versions of the GUI3DXBot software tool.

Table 2 shows the statistical survey results about GUI3DXBot. To validate them, it is expected that the maximum value deviation is less than 25 % as depicted in (1).

$$Cv = \frac{|5-\bar{x}|}{\bar{5}} \times 100 < 25\% \quad (1)$$

Where $\bar{x}$ is the arithmetic mean, which in our case it is desirable that $\bar{x} > 4$. In addition, it is expected that the mode corresponds to the maximum value (5). Then according to results in Table 2, the deviation coefficient (Cv) is less than 25 %, the mean is bigger than 4, and the mode is 5 in 63 % of these results.

Table 2. Statistical survey results of about GUI3DXBot

Question	Mode	f_{Mo} (%)	$\bar{x}$	Cv (%)
1	4	57.6	4.4	13.0
2	4	51.5	4.5	11.5
3	5	84.8	4.8	3.1
4	5	57.6	4.6	9.3
5	5	81.8	4.8	3.8
6	4	69.7	4.3	16.2
7	5	48.5	4.4	13.8
8	5	51.5	4.5	10.7
9	4	63.6	4.2	17.9
10	5	51.5	4.5	10.7
11	5	54.5	4.5	10.0

Source: The authors.

This work used an LRF as the main sensor just like 81 % of related works included in Table 1 that provided good results for safe robot navigation and people guidance. This sensor also allows creating an occupancy grid map just as 44 % of works in Table 1, which to our knowledge is a better human-robot interface to represent the environment. In addition, a Bayes-based framework was used in the manner of 75 % of works in Table 1, which in this case is a good choice given the dynamic environment faced by the mobile robot. Finally, the touchscreen which is the human-robot interface selected to capture the human-robot interaction was a wise choice, since according to results of Question 2, it obtained a mean of 4.5.-

Conclusion

This paper presented the development and testing of different software modules of the GUI3DXBot software tool, which offers an automated guidance service at the School of Electrical and Electronic Engineering at the Universidad del Valle. GUI3DXBot is a client-server application, in which the server-side is in charge of building a map of the environment, localizing the mobile robot,

planning the route to the user's goal, avoiding obstacles, always returning home at the end of the tour-guide service, and raising alerts in case of emergency. In turn, the client-side has the following functionalities: implementing the human-robot interaction, providing the goal list to users, displaying the local and global map view, receiving the user's selected goals, showing status information using a robot avatar, and giving emergency information.

The GUI3DXBot server side was developed using ROS, specifically the following packages: *RosAria* for robot motion and sensing, *GMapping* to build the environment map, *AMCL* for robot localization, *Navfn* for local and global planning, and web-socket support to interact with the client-side. The HRI was developed in Android using a 10-in tablet. Both server and client sides of GUI3DXBot were developed considering the RUP methodology. Fig. 2 illustrates the software module design which integrates all these packages to guide users in an office building.

To validate the functionality of GUI3DXBot a total of 33 human-robot interactions were documented using a survey. This survey aimed at evaluating the experience with the guidance service. As a result, users described GUI3DXBot as a *Very Nice* and *Intuitive GUI*, found it very easy to select goals, and could understand the interaction messages from GUI3DXBot. Ninety percent of users found the robot icon on the map and the local and global view of the environment map useful, as well as drawing the planned trajectory on the map since it helps them to orient themselves. Generally, the guidance service experience was very satisfactory (70 %) and satisfactory (30 %). This demonstrates that the robot system was well integrated and fully tested in real-world conditions.

Future works may include increasing the human-robot interaction by adding functionalities such as voice recognition and visual people recognition. Through these functionalities, the mobile robot can naturally interact with people and follow them in case they lose interest in the guidance service. Also, adding a redundant localization and mapping method based on features on the ceiling improve robot localization since LRF sensors cannot work properly in crowded environments.

References

[1] K.-T. Song, Y.-H. Chiu, S.-H. Song, and K. Zinchenko, "Scheduling and control of a cloud robot for reception and guidance," in 2017 International Automatic Control Conference (CACS), 2017, pp. 1-6. https://doi.org/10.1109/CACS.2017.8284263

[2] R. Triebel et al., "SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports," Springer, Cham, 2016, pp. 607-622. https://doi.org/10.1007/978-3-319-27702-8_40

[3] B. Bacca-Cortes, K. Muñoz Peña, and E. Caicedo Bravo, "GUI3DXBOT, Solución para la Asistencia a Peatones en Ambientes de Oficina Usando un Robot Móvil Guía (13-60-371)." Ministerio del Interior, Dirección Nacional de Derechos de Autor, Cali, 2017, p. 1.

[4] M. Quigley et al., "ROS: an open-source Robot Operating System," in ICRA, 2009, pp. 1-6.

[5] S. Thrun et al., "Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva," *Int. J. Rob. Res.*, vol. 19, no. 11, Nov., pp. 972-999, 2000. https://doi.org/10.1177/02783640022067922

[6] I. R. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto, "An affective mobile robot educator with a full-time job," *Artif. Intell.*, vol. 114, no. 1-2, Oct., pp. 95-124, 1999. https://doi.org/10.1016/S0004-3702(99)00027-2

[7] A. Bellarbi, S. Kahlouche, N. Achour, and N. Ouadah, "A social planning and navigation for tour-guide robot in human environment," in 2016 8th International Conference on Modelling, Identification and Control (ICMIC), 2016, pp. 622-627. https://doi.org/10.1109/ICMIC.2016.7804186

[8] S. J. Lee, J. Lim, G. Tewolde, and J. Kwon, "Autonomous tour guide robot by using ultrasonic range sensors and QR code recognition in indoor environment," in IEEE International Conference on Electro/Information Technology, 2014, pp. 410-415. https://doi.org/10.1109/EIT.2014.6871799

[9] S. Wang and H. I. Christensen, "TritonBot: First Lessons Learned from Deployment of a Long-Term Autonomy Tour Guide Robot," in 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2018, pp. 158-165. https://doi.org/10.1109/ROMAN.2018.8525845

[10] D. Rodriguez-Losada, F. Matia, R. Galan, M. Hernando, J. Manuel, and J. Manuel, "Urbano, an Interactive

Mobile Tour-Guide Robot," in Advances in Service Robotics, InTech, 2008. https://doi.org/10.5772/5950

[11] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "Autonomous Robot Navigation in Highly Populated Pedestrian Zones," J. F. Robot., vol. 32, no. 4, Jun., pp. 565-589, 2015. https://doi.org/10.1002/rob.21534

[12] K. Yelamarthi, S. Sherbrook, J. Beckwith, M. Williams, and R. Lefief, "An RFID based autonomous indoor tour guide robot," in 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), 2012, pp. 562-565. https://doi.org/10.1109/MWSCAS.2012.6292082

[13] C.-C. Tsai, S.-M. Shish, H.-C. Huang, M.-Y. Wang, and C. C. Lee, "Autonomous navigation of an indoor tour guide robot," in 2008 IEEE Workshop on Advanced robotics and Its Social Impacts, 2008, pp. 1-6. https://doi.org/10.1109/ARSO.2008.4653591

[14] F. Faber et al., "The humanoid museum tour guide Robotinho," in RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication, 2009, pp. 891-896. https://doi.org/10.1109/ROMAN.2009.5326326

[15] H.-M. Gross et al., "TOOMAS: Interactive Shopping Guide robots in everyday use - final implementation and experiences from long-term field trials," in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 2005-2012. https://doi.org/10.1109/IROS.2009.5354497

[16] Y. Chen, F. Wu, W. Shuai, and X. Chen, "Robots serve humans in public places-KeJia robot as a shopping assistant," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 3, pp. 1-20, 2017. https://doi.org/10.1177/1729881417703569

[17] R. Stricker et al., "Konrad and Suse, two robots guiding visitors in a university building," *Inform. aktuell*, pp. 49-58, 2012. https://doi.org/10.1007/978-3-642-32217-4_6

[18] H. Lin and W. Tsao, "Automatic mapping and localization of a tour guide robot by fusing active RFID and ranging laser scanner," in 2011 Int. Conf. Adv. Mechatron. Syst., pp. 429-434, 2011.

[19] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Trans. Robot.*, vol. 23, no. 1, Feb., pp. 34-46, 2007. https://doi.org/10.1109/TRO.2006.889486

[20] P. Kruchten, The Rational Unified Process: An Introduction, 3rd ed. Addison-Wesley Professional, 2003.

[21] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 709-715.

[22] J. Mace, J. Lee, R. Toris, B. Alexander, D. Bertram, and M. Gruhler, "RosBridge Protocol Specification," GitHub, 2018. [Online]. Available: https://github.com/RobotWebTools/rosbridge_suite/blob/groovy-devel/ROSBRIDGE_PROTOCOL.md. [Accessed September 1, 2018].