Research Article

Sébastien Laniel, Dominic Létourneau, François Grondin, Mathieu Labbé, François Ferland, and François Michaud*

Toward enhancing the autonomy of a telepresence mobile robot for remote home care assistance

https://doi.org/10.1515/pjbr-2021-0016
received July 25, 2019; accepted March 2, 2021

Abstract: In health care, a telepresence robot could be used to have a clinician or a caregiver assist seniors in their homes, without having to travel to these locations. However, the usability of these platforms for such applications requires that they can navigate and interact with a certain level of autonomy. For instance, robots should be able to go to their charging station in case of low energy level or telecommunication failure. The remote operator could be assisted by the robot’s capabilities to navigate safely at home and to follow and track people with whom to interact. This requires the integration of autonomous decision-making capabilities on a platform equipped with appropriate sensing and action modalities, which are validated out in the laboratory and in real homes. To document and study these translational issues, this article presents such integration on a Beam telepresence platform using three open-source libraries for integrated robot control architecture, autonomous navigation and sound processing, developed with real-time, limited processing and robustness requirements, so that they can work in real-life settings. Validation of the resulting platform, named SAM, is presented based on the trials carried out in 10 homes. Observations made provide guidance on what to improve and will help identify interaction scenarios for the upcoming usability studies with seniors, clinicians and caregivers.

Keywords: mobile robot assistant, assisted telepresence, autonomous navigation, voice tracking, robot control architecture, evaluation in home settings

1 Introduction

Around the world, problems caused by population aging drive interest in developing new technology, including robotics [1,2], to provide home care. Telehomecare, or home telehealth, consists of providing health care services into a patient’s home [3] and is certainly an area of interest for telepresence mobile robots [4–9] in tasks such as telehomecare visits, vital sign monitoring and Activity of Daily Living (ADL) assistance [9] for instance.

Mobile telepresence robotic platforms usually consist of a mobile base, a camera, a screen, loudspeakers and a microphone, making them mobile videoconference systems, commonly referred by some to be “Skype on wheels” [10]. Commercial consumer-based mobile telepresence robotic platforms have been available over the last decade (see reviews in ref. [4,8,11–16]) and provide mobility to sensors, effectors and interactive devices for usage in hospitals, offices and homes [17], outlining recommendations for moving toward their use in practical settings. Most have no or very limited autonomy [4,8,18] which, according to ref. [4], is attributed to simplicity, scalability and affordability reasons. For telehomecare applications, the remote operator, who would most likely be novice robot users (e.g., clinicians, caregivers), would find it beneficial to receive assistance in navigating in the operating environment [4,16,19] and in following and tracking people (visually and from voice localization) with whom to interact [4,18]. Such capabilities would minimize what the remote operators have to do to control the platform and to focus on the interaction tasks to be conducted through telepresence [4,20].

In addition, most of the work on telepresence are not at all evaluated in real environments [21–24], nor do they underline the difficulties encountered and the limitations of their designs [1,2]. Autonomous capabilities can work well in lab conditions but may still have limitations when deployed in home environments, making it important to
conduct trials in such conditions to move toward the use of telepresence mobile robots for remote home care assistance [4].

Addressing the issues of autonomy and trials in real home environments with a telepresence mobile robot requires to have access to such a platform along with the targeted autonomous capabilities. This article presents how we try to address these design considerations by developing SAM [8], an augmented telepresence robot from Suitable Technologies Inc. programmed using a robot control architecture with navigation and sound processing capabilities. To benefit from the progress made in these areas and to be able to focus on the integration challenge in designing a robot for remote home care assistance, we use for convenience open-source libraries we designed and used by the research community. These libraries are designed with online processing and real-world constraints using robots with limited online processing capabilities in mind; and by being open-source, they provide replicability of the implementation for experimental purposes. Results from trials conducted in 10 home environments (apartments, houses and senior residences) are presented. Autonomous navigation capabilities in reaching waypoints or going back to the charging station are evaluated by navigating inside rooms or to different rooms. Autonomous conversation following in quiet and noisy conditions is also evaluated. The purpose of these trials is to assess SAM in real home settings before conducting usability studies, to determine the improvements to be made and under which conditions its autonomous capabilities can be used.

The article is organized as follows. First, Section 2 presents related work on telepresence robots for home care along with the design choices we made for the robot platform, the robot control architecture and the navigation and sound processing capabilities. Sections 3 and 4 present SAM hardware and control implementation, respectively. Section 5 describes the experimental methodology used to test SAM’s autonomous capabilities in home settings, followed by Section 6 with results and observations. Section 7 presents the limitations of the work reported, with Section 8 concludes the article.

2 Related work and design choices

To our knowledge, the ExCITE (Enabling SoCial Inter-action Through Embodiment) project [4,25–28] using the Giraff telepresence robot is the only one that addresses telepresence in the context of home care. It presents very interesting and detailed methodologies, observations and requirements for moving forward with deploying telepresence mobile robots for remote home care assistance. The Giraff robot platform is a closed system being available only for purchase within Sweden at a cost of $11,900 USD.¹ It has a zoom camera with a wide-angle lens, one microphone, one speaker, a 13.3” LCD screen mounted on top of a base and a charging station to charge its battery [29]. The Giraff robot is used to provide security, medical follow-up and assistance to daily activities of seniors. It has its own middleware for interfacing sensors [29,30]. Home sensors, medical sensors and the Giraff robot are connected to a cloud-based system to retrieve the information taken by the various sensors to monitor the patient’s activities, e.g., evaluates the daily time spent sitting on a chair, detects in which room the elder is, monitors weight, blood pressure and blood glucose levels. Short-term and long-term studies over 42 months and 21 test sites in three European countries are reported [4,25,28], along with insightful quantitative and qualitative research methodologies of user needs and validation [20,26,27] and design recommendations [4].

One of such recommendation is as follows: “Developers of MRP system for use in homes of elderly people should strive to provide obstacle detection, and a map indicating the position of the robot, to ease the docking procedure” [4]. This is an essential feature to avoid having the remote operator teleoperate the robot back to its charging station at the end of a session, or to be moved out of the way by the occupant in case of low energy level or a telecommunication failure [4]. It requires the robot to have mapping and localization capabilities, allowing it to navigate efficiently and safely in the home. High-level descriptions of autonomous capabilities integrated for safe navigation (using a 2D laser range finder and a camera) and user interfaces are provided [4]. However, they are insufficient to reimplement them and their performance remain uncharacterized. To build from the findings reported in the ExCITE project and provide additional contributions regarding autonomous capabilities requires having access to a telepresence development platform. To provide a foundation on which to build on, we decided to focus on three components related to autonomy: robot control architecture, autonomous navigation and sound processing capabilities. Each autonomous capability brings its share of individual and integration challenges [31] and is a research endeavour on its own. Because providing detailed reviews of the state of the art in each of these areas is outside the scope of the article,
the following subsections situate and explain the design choices we made to implement SAM and the targeted capabilities using our own libraries.

### 2.1 Robot platform

For home care, telepresence robots should be lightweight to facilitate their installation and their manipulation, stable to avoid potential hazard in case of hardware failure or physical contacts, and inexpensive. When we started this project in 2015, we first conducted a review of the different telepresence platforms to determine whether we needed to design our own or simply use one available in the market. Most platforms use differential drive locomotion with some being self-balanced using only two wheels, making them unstable if someone tries to lean onto it. Omnidirectional locomotion facilitates navigation in tight spaces, which could be quite useful in homes if the cost of the platform remains low. UBBO Maker robot [32] has such capability, but has limited payload to add sensors for autonomous navigation or vital sign monitoring. Based on these observations, we chose to use the Beam platform. At the time, it was one of the least expensive platform (USD 2,000). It can be interfaced with the library presented in ref. [33] to control the motors with velocity commands and to read odometry.

### 2.2 Robot control architecture

Providing more decisional autonomy to robots requires the use of a robot control architecture. Robot control architectures define the interrelations between decision-making modules required by the application. With continuous technological progress and availability of higher processing and interacting capabilities, robot control integration framework (a.k.a. architecture) facilitates expandability and portability. There is an infinite number of ways to implement robot control architectures (see review in ref. [34]), making it hard to compare them [35] because research on robot control architectures is conducted more as feasibility-type studies. For instance, designing robot control architectures is being addressed in robot competitions such as the RoboCup@HOME, aiming to develop service and assistive robot technology with high relevance for future personal domestic applications. A frequently used control architecture is the layered, or tiered, robot control architecture, with layers usually organized according to the principle of increasing precision with decreasing intelligence [36]. The most common robot control architecture used in this context has three layers: deliberative (high level, abstract reasoning and task planning), executive (task coordination) and functional (task execution). For instance, the Donaxi robot [37,38] has a deliberative layer (for symbolic representation and reasoning), an executive layer (for plan monitoring) and a functional layer. Siepmann et al. [39] uses a hardware layer, a functional layer and a BonSAI layer. The complexity in layered robot control architecture comes in how to interface and partition these layers [40]. Although there is no consensus on a common architecture, how to engineer a system that effectively integrates the functionalities required is an open question of fundamental importance in robotics [41], and there is currently no dominant solution [42].

In our case, we use HBBA (Hybrid Behavior-Based Architecture) [43,44], an open source² and unifying framework for integrated design of autonomous robots. Illustrated Figure 1, HBBA is a behavior-based architecture with no central representation that provides the possibility of high-level modeling, reasoning and planning capabilities through Motivation or Perception modules. Basically, it allows Behaviors to be configured and activated according to what are referred to as the Intentions of the robot. Intentions are data structures providing the configuration and activation of Behaviors (i.e., the behavioral strategy) and the modulation of Perception modules. As the number and complexity of Perception modules, Behaviors and Motivations increase to address more sophisticated interaction scenarios, the Intention Workspace becomes critical. While layered architectures usually impose a specific deliberative structure (for instance a task planner) to coordinate the lower-level Behaviors, HBBA can use multiple concurrent independent modules at its highest level, without constraining those modules to a specific decisional scheme. Compared to more formal planning approaches such as Konidaris and Hayes [45], HBBA is a robot control architecture presenting design guidelines and working principles for the different processing modules, without imposing a formal coding structure for its implementation. HBBA’s generic coordination mechanism of Behaviors has demonstrated its ability to address a wide range of cognitive capabilities, ranging from assisted teleoperation to selective attention and episodic memory, simply by coordinating the activation and configuration of perception and behavior modules. It has also been used with humanoid robots

---

² http://github.com/francoisferland/hbba
such as the NAO and Meka Robotics M1 in an episodic memory sharing setup [46], and with the Robosoft Kompai and later on the PAL Robotics TIAGo as service robots for the elderly with mild cognitive impairments [47].

### 2.3 Autonomous navigation

SPLAM (Simultaneous Planning, Localization And Mapping) [48] is the ability to simultaneously map an environment, localize itself in it and plan paths using this information. This task can be particularly complex when done online by a robot with limited computing resources. A key feature in SPLAM is detecting previously visited areas to reduce map errors, a process known as loop closure detection. For usage in home settings, the robot must be able to deal with the so-called kidnapped robot problem and the initial state problem: when it is turned on, a robot does not know its relative position to a map previously created, and it has, on startup, to initialize a new map with its own referential; when a previously visited location is encountered, the transformation between the two maps can be computed. Appearance-based loop closure detection approaches exploit the distinctiveness of images by comparing previous images with the current one. When loop closures are found between the maps, a global graph can be created by combining the maps into one. However, for large-scale and long-term operation, the bigger the map is, the higher is the computing power required to process the data online if all the images gathered are examined. With limited computing resources on mobile robots, online map updating is limited, and so some parts of the map must be somewhat forgotten.

Memory management approaches can be used to limit the size of the map, so that loop closure detection is always processed under a fixed time limit, thus satisfying online requirements for long-term and large-scale environment mapping. RTAB-Map (Real-Time Appearance-Based Mapping)³ [49–51] is an open-source library implementing such an approach, using images of the operating environment. Being visual based, RTAB-Map can also provide 3D visualization of the operating environment from video data, which may assist the remote user in navigation tasks [4]. Released in 2013, RTAB-Map can be used as a cross-platform standalone C++ library and with its ROS package⁴ to do 2D or 3D SLAM.

Figure 2 illustrates an example of a 3D and a 2D map representations created with RTAB-Map using a Kinect camera and a 2D LiDAR. These representations can be useful to assist the remote operator, in particular the 3D representation [4]. The Kinect camera generates a depth image coupled with a standard RGB image, resulting in a colored 3D point cloud. The RGB image is also used to calculate image features stored in a database. RTAB-Map combines multiple point clouds together with transforms (3D rotations and translations) from one point cloud to the next. Estimation of the transforms are calculated from the robot’s odometry using wheel encoders, visual odometry or sensor fusion [52]. Image features from the current image are compared to the previously calculated image features in the database. When the features have a strong correlation, a loop closure is detected. Accumulated errors in the map can then be minimized using the new constraint leading to a corrected map [53]. As the map increases in size, loop closure detection and graph optimization take more and more processing time. But RTAB-Map’s memory management approach transfers, when a fixed real-time limit is reached, i.e., oldest and less seen locations into a long-term memory where they are not used for loop closure detection and graph optimization, thus bounding the map update time to a deter-

---

³ http://introlab.github.io/rtabmap
⁴ http://wiki.ros.org/rtabmap_ros
mined threshold. When a loop closure is found with an old location still in working memory, its neighbor locations are brought back from the long-term memory to the working memory for additional loop closure detection and to extend the current local map.

2.4 Sound processing

Robots for home assistance have to operate in noisy environments, and limitations are observed in such conditions when using only one or two microphones [54]. For instance, sound source localization could be used for localizing the resident [4] or localizing the speaker when engaged in conversation with several users [18]. A microphone array can enhance performance by allowing a robot to localize, track and separate multiple sound sources to improve situation awareness and user experience [18]. Sound processing capability, combined with face tracking capabilities, can be used to facilitate localization of the occupants [4] and to position the robot when conversing with one or multiple people in the room [4,16,18,19], again to facilitate the task of navigating the platform by allowing the remote operator to focus on the interaction with people.

ODAS [55] is an open-source library5 performing sound sources localization, tracking and separation. Figure 3 shows the main components of the ODAS framework. ODAS improves robustness to noise by increasing the number of microphones used while reducing computational load. This library relies on a localization method called Steered Response Power with Phase Transform based on Hierarchical Search with Directivity model and Automatic calibration (SRP-PHAT-HSDA). Localization generates noisy potential sources, which are then filtered with a tracking method based on a modified 3D Kalman filter (M3K) that generates one or many tracked sources. The module’s output can be used to continuously orient the robot’s heading in the speaker’s direction, and sound locations can be displayed on the remote operator 3D interface [56]. Sound sources are then filtered and separated using directive geometric source separation (DGSS) to focus the robot’s attention only on speech, and ignore ambient noise. The ODAS library also models microphones as sensors with a directive polar pattern, which improves sound sources localization, tracking and separation when the direct path between microphones and the sound sources is obstructed by the robot’s body.

To make use of ODAS, a sound card and microphones are required. Commercial sound cards present limitations when used for embedded robotic applications: they are usually expensive; they have functionalities unnecessary for robot sound processing and they also require significant amount of power and size. To facilitate the use of ODAS on various robotic platforms, we also provide as open hardware two sound cards [57]: 8SoundsUSB6 and 16SoundsUSB,7 for 8 and 16 microphone arrays, respectively. They provide synchronous acquisition of microphone signals through USB to the robot’s computer.

3 SAM, a remote-assistance robot platform

A standard Beam platform comes with a 10″ LCD screen, low-power embedded computer, two 640 × 480 HDR (High Dynamic Range) wide-angle cameras facing bottom and front, loudspeakers, four high-quality microphones, WiFi network adapter, a 20-AH sealed lead-acid 12 V battery capable of approximately 2 hours of autonomy. It also comes with a charging station: the operator just has to position the robot in front of it and activate the docking mode to let the robot turn and back up on the charging station. The robot's

5 http://odas.io
6 https://sourceforge.net/projects/eightsoundsusb/
7 https://github.com/introlab/16SoundsUSB
dimensions are 134.4 cm (H) × 31.3 cm (W) × 41.7 cm (D). The platform also uses wheel encoders and an inertial measurement unit to estimate the change in position of the robot over time. Motor control and power management are accomplished via an USB 2.0 controller in the robot’s base, and its maximum speed is 0.45 m/s.

As shown by Figure 4, we placed a Kinect camera on top of the LCD screen using custom-made aluminum brackets, facing forward and slightly inclined to the ground. Considering the Kinect’s limited field of view, placing the Kinect on top of the robot makes it possible to prevent hitting hanging objects or elevated shelves and to perceive objects on tables or counters. We installed a circular microphone array using an 8SoundsUSB [57] sound card and customized aluminum brackets and acrylic support plates at 67 cm from the ground. We added an Intel Skull Canyon NUC6i7KYK (NUC) computer equipped with a 512-GB hard drive, 32 GB RAM, a quad Core-i7 processor, USB3 ports, Ethernet and WiFi networking. We replaced the head computer’s hard drive with a 128-GB mSATA drive. Both computers run Ubuntu 16.04 operating system with ROS (Robot Operating Systems) [58] Kinetic. We electrically separated the added components and the original robot by using SWX HyperCore 98Wh V-Mount-certified lithium-ion battery (protected in over/under-voltage and current) placed on the robot’s base using a V-Mount battery plate, keeping the robot’s center of gravity as low as possible and facilitating battery swapping for charging. Using an additional battery is not ideal because it complexifies the charging process of the robot, limiting its use to trained users. However, this allows us to revert any changes and to keep our modifications as less intrusive as possible. Coupled with DC–DC converters, the battery provides power to the microphone array, the Kinect and the NUC computer. The lithium-ion battery is recharged manually and separately. This configuration gives 50 minutes of autonomy when the robot maps its environment, and 75 minutes when using navigation modalities (i.e., autonomous navigation, teleoperation). Overall, the additional components plus the initial robot platform USD 4,300.

Telepresence robots used for health-care applications, such as RP-VITA [59] and Giraff [60], interface with vital sign monitoring devices for medical follow-up. To implement such capabilities, a low-cost USB dongle is installed on SAM to acquire the following vital signs from battery-powered Bluetooth Low Energy (BLE) sensors: blood pressure, SPO2 and heart rate, temperature, weight scale and glucometer [61]. In our case, we also design our own telecommunication framework for tele-health applications [61], addressing the needs of remote home care assistance applications.

4 SAM’s robot control architecture

Figure 5 illustrates the implementation of SAM’s robot control architecture, following the HBBA framework, to make SAM a remote home care assistance robot. As a general overview, its main motivations are Survive and Assistive Teleoperation. Survive supervises the battery level and generates a Desire to go to the charging station when the battery level is too low. Using the interface, the remote operator can activate autonomous functionalities managed by Assistive Teleoperation. This allows the user to either manually control the robot, to communicate a high level destination for autonomous navigation, to autonomously track a face or autonomously orient SAM toward a person talking. The following sections provide more details on the Sensors, Perception, Behaviors, Actuators and Motivations modules implemented for SAM.

4.1 Sensors

SAM has the following input sensory modules as shown in Figure 5.
• **Battery Level** monitors the battery voltage level and current consumption in floating point units.

• **Gamepad** is a wireless controller shown in Figure 6 and used to activate or deactivate the wheel motors. It allows the operator to manually navigate the robot or to activate SAM’s autonomous modalities.

• **Operator GUI** (Graphical User Interface) shown in Figure 7 allows the operator to teleoperate SAM and to activate SAM’s autonomous modalities.

• **Kinect** is the RGB-D data generated by the Kinect camera.

• **Floor Camera** is a webcam facing the ground and used to locate the charging station.

• **Microphone Array** is the 8-microphone array installed on SAM.

• **Odometry** is data provided by wheel encoders and the inertial measurement unit of the Beam platform to estimate its change in position over time.

• **Head Camera** is the webcam installed in SAM’s forehead, facing forward, for visual interaction with people.

• **Wireless Vital Sign Sensors** is the BLE interface for the wireless vital signs monitoring devices.

### 4.1.1 Perception

The Perception modules process Sensors data into useful information for the Behaviors. SAM’s Perception modules shown in Figure 5 are:

- **RTAB-Map** for mapping and localization, as presented in Section 2.3.

- **Symbol Recognition** uses **Floor Camera** to detect the symbol on the charging station, shown in Figure 8, and identifies its orientation using ROS find_object_2d package.8 This package uses OpenCV9 to detect the charging station. The homography between the corresponding features of the reference image and the scene image is computed. Three points are chosen in the reference image to estimate the pose of the object in the scene image: one directly in the middle of the object, one along the x axis of the object and one along the y axis of the object. Knowing the angle of the Floor Camera, the position (x, y) relative to the robot is calculated.

- **ODAS** is for sound source localization, tracking and separation, as explained in Section 2.4.

---

8 [http://wiki.ros.org/find_object_2d](http://wiki.ros.org/find_object_2d)

9 [https://opencv.org/](https://opencv.org/)
Face Recognition uses SAM’s Head Camera in conjunction with FisherFaceRecognizer in openCV 3.0 library. It compares the actual face with all the prerecorded faces in its database. Then, Face Recognition identifies the most likely person with a confidence score.

VSM (Vital Sign Monitoring) translates vital sign data into a generic JSON format, to be compatible with software used by Show and Log.

### 4.1.2 Behaviors

SAM’s Behaviors, illustrated in Figure 5 and designed by us, are control modalities organized with a priority-based action selection scheme as follows:

- **Manual Teleoperation** is the highest priority Behavior, giving absolute control to an operator using the Gamepad. This Behavior is used for security interventions and during a mapping session.
- **Obstacle Avoidance** plans a path around an obstacle detected in the robot’s the local map, to avoid collisions.
- **Go To** allows SAM to navigate autonomously using SPLAM provided by the RTAB-Map module.
- **Dock** allows the robot to connect itself to the charging station when it is detected. As shown in Figure 8, the charging station has a flat part on which to roll over, with a symbol used to calculate the distance and orientation of the charging station. With the Beam head’s hard drive replaced, we could not interface this behavior with the existing docking algorithm of the Beam platform. Therefore, we had to implement our own. The robot charging connector is at the back of its base and there is no sensor to navigate backward. Therefore, before turning to dock backward, the robot must generate a path. Shown in Figure 9, our algorithm uses a \((x, y)\) representation centered on the charging station at \((0, 0)\), pointing to the left \((-x)\) and the robot position at \((x_i, y_i)\). Symbol Recognition gives the position \((x_R^S, y_R^S)\) and orientation \(\theta_R^S\) of the charging station relative to the robot. Then the robot position relative to the charging station \((x, y)\) is found:

\[
\begin{align*}
  x &= -x_R^S \cos \theta_R^S - y_R^S \sin \theta_R^S \\
  y &= x_R^S \sin \theta_R^S - y_R^S \cos \theta_R^S
\end{align*}
\] (1)

To connect the robot perpendicularly to the charging station, a second-order polynomial path is chosen:

\[
y = \frac{y_1}{x_1^2} x^2
\] (3)

Once the path is calculated, the initial orientation \(\theta_i\) of the robot is found using the derivative of (3). The robot turns in place to reach \(\theta_i\). It then starts to move backwards following the path. To monitor the movement, Odometry provides the robot position \((x_M, y_M)\) and orientation \((\theta_M^A)\) with respect to the map. This position in relation to the charging station \((x_A, y_A)\) is found by applying the homogeneous matrices \((A_M^M, A_S^M)\) that change the coordinate system from map to robot and from robot to charging station, respectively.

\[
\begin{bmatrix}
x_A \\
y_A \\
1
\end{bmatrix} = A_M^M A_S^R
\begin{bmatrix}
x_M^R \\
y_M^R \\
1
\end{bmatrix}
\] (4)

Using a cycling rate of 100 Hz, the velocities are defined by a translational velocity of 0.4 m/s and a rotational velocity defined by:

\[
\dot{\theta}_{\text{twist}} = (\theta_A^S - \theta_i) \times 100
\] (5)

When Odometry indicates that the robot is not moving and that there is indeed a non-zero speed command sent to the base, it means that the robot encountered
an obstacle, potentially the charging station. It then stops for 1 s; and if the battery’s current consumption becomes negative within this period, the robot is docked and charged. If not, the robot continues to back up according to the calculated trajectory.

- **Voice Following** uses ODAS to perceive multiple sound source locations, amplitudes and types (voice or non-voice). The main interlocutor is considered to be the voice source with the highest energy. Its location and the robot’s odometry are used to turn and face the main interlocutor.

- **Face Following** follows the closest face detected by Face Recognition using the TLD Predator (Tracking Learning and Detection) [62] package. As shown by Figure 10, once a face is detected, Face Following is able to track it even if it becomes covered or it changes orientation. The current implementation only tracks one face at a time.

- **Speak** converts predefined texts into speech using Festival speech synthesis and the sound_play ROS package.¹

- **Show** displays, on the robot’s screen, the remote operator webcam, vital signs and the robot’s battery level, as shown by Figure 7.

### 4.1.3 Actuators

The Action Selection module receives all the Actions generated by the activated Behaviors and keeps the ones from the highest priority Behaviors for the same Actuator. Actuators shown in Figure 5 are:

- **Base** translates velocity commands into control data for the wheel motors.
- **Voice** plays sounds coming from Speak or the audio coming from the operator’s voice.
- **Screen** displays the info from Show.
- **Log** saves all vital signs gathered from VSM into a Firebase database, a Google web application.¹¹ Data are logged with a time stamp.

### 4.1.4 Motivations

SAM’s Motivations are:

- **Survive** monitors SAM’s Battery Level and generates a Desire to return to the charging station when battery voltage is lower than 11.5 V.
- **Assistive Teleoperation** allows the remote operator, using the GUI, to activate autonomous modes for navigation, for following a conversation or for following a person’s face. By default, when no signals are coming from Operator GUI, a Desire to return to the charging station is generated.

### 4.2 Validation in lab conditions

SAM’s functionalities allows the operator to map the environment, to navigate autonomously in the resulting map and dock into its charging station, to let the robot position itself in the direction of the person talking and to track a person by following a face. These functionalities can be individually activated using Operator GUI. Before conducting trials in real homes, we validated their efficiency and reliability in our lab facility. After having created a reference map of hallways and rooms using RTAB-Map, autonomous navigation was tested by having SAM move to different goal point locations. The robot safely moved in hallways, around people, around furniture (workbenches, tables, chairs, equipment of various types) and through door frames. To emulate home-like conditions, the lab’s door frame was narrowed to 71 cm using a plywood. The charging station was placed against a wall in an open area to validate the motivation Survive, i.e., making the robot return to the charging station. This function was successfully validated over traveling distance ranging from 1 to 20 m. Autonomous conversation following was tested in different rooms and during public demonstrations. Face recognition was validated with different participants individually, also in different rooms. These trials done in controlled conditions were all successful, suggesting that SAM was ready to be tested in more open and diverse experimental conditions.

### 5 Experimental methodology

As stated in the introduction, the objective is to examine the efficiency and reliability of SAM’s modalities in real
home settings. In each new home setting, the first step involved positioning the charging station against a wall in an area with enough space (1 m²) for the robot to turn and dock. Second, every door frame width, door threshold height and hallway width were manually measured using a tape measure, to characterize the environments and provide observations when SAM experienced difficulties in these areas. Environment limitations were also identified, specifically stairs, steps (≥0.5 cm) and rooms forbidden by the residents. Third, an operator created a reference map using the Gamepad and Manual Teleoperation and by navigating in the different rooms, making sure to fully map the walls and furniture by looking at RTAB-Map’s reference map displayed using rviz. If the operator found the map to be an adequate representation of the home, he then identified the locations for the Go To behavior on the map. Since this article aimed to examine the efficiency and reliability of SAM’s modalities in real home settings, For consistency, the experiments were conducted by the same person, experienced in operating SAM.

Early on, as we followed this process, we noticed that some adjustments were required regarding SAM’s configuration and usage compared to its validation in laboratory conditions:

- The position of the Kinect camera brings limitations for mapping and navigating. Illustrated in Figure 11, the Kinect’s vertical field of view (FOV) of 60° creates a blind spot. The blind spot causes misinterpretations when approaching obstacles, like chairs and tables. To limit this, the operator made the robot stay at least 40 cm away from obstacles that could be partially seen because of the blind spot. In addition, robot’s accelerations, floor slope and floor cracks generate vibrations. And as shown in Figure 12, a change of 2° of the Kinect’s orientation can cause misinterpretation errors, for instance, registering the floor as an obstacle. To prevent this, we set the minimum obstacle height at 20 cm.

- The Kinect camera has problems sensing mirrors or reflective objects: all obstacles reflected by a mirror are seen as through a window. This adds noise or ghost obstacles. We tried to minimize this effect by first mapping rooms with large mirrors and then proceed with the other rooms, attempting to remove noise and ghost obstacles.

- Difficulties were noticed with Face Following. With the change in brightness level between each room and with time of day, Face Following revealed to be unreliable in real-life conditions while it performed well in the lab. We therefore decided to leave this functionality out of the experiments, to focus on autonomous navigation and autonomous conversation following.

5.1 Autonomous navigation

An autonomous navigation trial involves having SAM move from an initial to a goal location, which is referred to as a path. For each trial, the operator used the Operator GUI to chose between the Go To behavior to go to a pre-defined location or the Return to the charging station behavior to have SAM go to its charging station. As the robot moved during a trial, the operator held the enable button on the Gamepad and looked at RTAB-Map’s reference map and the video feeds from both the Head Camera and the Floor Camera, releasing the enable button to intervene when necessary. Since SAM is a telepresence robot, we consider such intervention acceptable to compensate for the robot’s limitations. The operator then used Manual Teleoperation behavior to reposition the robot. For additional safety purposes, another person was also physically following SAM, ready to intervene if necessary. A trial is considered successful when the robot reaches its goal and the operator intervenes at most once to recover from the following types of cases:

![Figure 11: Kinect’s blind spot on SAM. Light gray area is interpreted as safe zone for navigation, and dark gray as obstacle. The chair is seen here as a smaller obstacle because the blind spot covers part of it.](image)

![Figure 12: Misinterpretation error caused by vibrations on the Kinect camera. Floor at 6 m of the Kinect is detected 20 cm higher if the pitch increases by 2°.](image)
• Avoid a collision by changing the robot’s orientation to move away from the obstacle.
• Overcome a path planning failure.
• Reposition the robot if the charging station is not visible from the Floor Camera or if the docking attempt was unsuccessful.

Depending on path taken from the initial and goal locations, trials were conducted, in no particular order, in the following four navigation scenarios:
1. Navigate in a room: the robot receives a destination and creates a path between its initial and final positions, without having to cross a hallway or a door frame.
2. Navigate to a different room: the robot has to move through door frames and hallways, making it possible to assess the impact of door frame sizes and hallway width during navigation.
3. Return to the charging station located in the same room: this involves to autonomously navigate and dock into the charging station located in the same room.
4. Return to the charging station located in a different room: same task but having the robot go through one or multiple door frames and hallways.

For each trial, the robot’s path, the time elapsed and the distance travelled were recorded. The type and number of operator interventions were also noted, along with observations during door frame crossing.

During mapping, observations were made regarding the creation of the reference map in real homes with SAM. First, the position of the Kinect camera brings limitations for mapping and navigating. Illustrated in Figure 11, the Kinect’s vertical field of view (FOV) of 60° creates a blind spot. The blind spot causes misinterpretations when approaching obstacles, like chairs and tables. To limit this, the operator made the robot stay at least 40 cm away from obstacles that could be partially seen because of the blind spot. In addition, robot’s accelerations, floor slope and floor cracks generate vibrations. And as shown in Figure 12, a change of 2° of the Kinect’s orientation can cause misinterpretation errors, for instance, registering the floor as an obstacle. To prevent this, we set the minimum obstacle height at 20 cm.

Second, the Kinect camera has problems sensing mirrors or reflective objects: all obstacles reflected by a mirror are seen as through a window. This adds noise or ghost obstacles. We tried to minimize this effect by first mapping rooms with large mirrors and then proceed with the other rooms, attempting to remove noise and ghost obstacles.

Third, the robot’s odometry influences navigation performance. SAM’s Odometry is calculated by the Beam base using wheel encoders and an inertial measurement unit. Rotation error is around 2.8% and linear error is roughly 0.8%. For each rotation in place, Odometry accumulates an error of up to 10°, which decreases the quality of the map derived by RTAB-Map.

Lastly, when mapping, RTAB-Map memorizes images with their visual features as references for loop closure. Loop closure occurs when a match is found between the current image and an image in memory, using similarity measures based on visual features in the images. One limitation is that every feature is assumed to be static and significant. This turned out to be problematic for autonomous navigation in laundry rooms and kitchens. For example, the top room of Environment B in Figure 17a is a laundry room. The first time the room was mapped, it had colorful clothes folded on the ironing table. The next day, RTAB-Map was unable to perform loop closure because the clothes were gone and the colorful features were not visible. This problem can occur in all kinds of context in the homes like mapping dishes, food, shoes, clothes, pets, chairs, plants or even doors (opened or closed). When RTAB-Map is unable to perform loop closure, the odometry error accumulates and the local map drifts from the global map. If the drift becomes too large, RTAB-Map is unable to find a possible path to both satisfy the local map and the global map, making autonomous navigation impossible. In this situation, the operator has to intervene and manually navigate the robot until RTAB-Map can perform a loop closure, resynchronizing SAM’s position in the map.

5.2 Autonomous conversation following

Autonomous conversation following aims to enhance the operator experience by autonomously directing the camera toward the person talking. Since face following isn’t reliable, we only test the Voice Following behavior. To provide repeatable experimental conditions, a pre-recorded audio conversation between two men was played from two speakers. Shown in Figure 13, the speakers were placed at different heights (43 cm to 1.4 m), angles (120° to 150°) and distances (1 m to 1.6 m) in environments A, B, E and J. The operator enabled the Voice Following behavior using the Operator GUI and played the pre-recorded
conversation, during which the active speaker changed 12 times. The operator observed and noted whether the robot was able to orient itself toward the active speaker when more than four syllables were heard.

Tests were conducted in two conditions:
• Quiet: no loud interference was heard throughout the conversation.
• Noisy: having typical sounds occurring in the home. For instance, home residents were told to resume their normal activities and therefore could watch television, listen to music, prepare meals, vacuum, etc., in addition to having regular home noise (e.g., kitchen hood, fan).

6 Results and observations

Table 4 of Appendix A presents the 10 different home settings where we conducted trials. They include a variety of different types of rooms, floor types, door widths and hallways, and configurations of various furniture. None were modified or adjusted to help the robot, except for doors that were either fully opened or closed. The sketches provided in Appendix A are approximate representations of the real homes. Examples of reference maps generated by RTAB-Map are also provided. The dark lines in the maps are obstacles and the gray areas are the safe zones for navigation. Home availability for experimentation ranged from 2 hours to 2 weeks.

6.1 Autonomous navigation

Depending on the availability and complexity of the home, one to five reference maps were created for each of the 10 home environments, for a total of 35 reference maps. For each reference map, two to six paths were tested, with each path repeated for at least three trials. Overall, 400 autonomous navigation trials were conducted.

As shown in Figure 14, trials lasted between 14 and 158 s, with an average of 38.5 s and a standard deviation of 19.5 s. Trials done autonomously lasted between 14 and 87 s, with an average of 30 s and a standard deviation of 11.1 s. Trials involving interventions from the operator lasted between 17 and 158 s, with an average of 53.7 s and a standard deviation of 24 s. Distances travelled are between 3.4 and 11.9 m, with an average of 6.9 m and a standard deviation of 2.3 m.

Table 1 presents results of the trials in relation to the four autonomous navigation scenarios defined in Section 5.1. Allowing the operator to intervene once (as explained in Section 5.1) led to 80 additional successful trials (368) compared to trials completed autonomously (288). Only 32 trials (400 minus 368, about 8.0%) were unsuccessful. When (1) Navigating in a room or (2) Navigating to a different room, SAM succeeded in 264 trials (94 + 170) over 270 trials (97 + 173) with intervention (i.e., 264 over 270 giving 97.8%), with 76.7% (81 + 126 = 207) autonomously. In these successful trials, the operator intervened 70 times (11 + 5 + 36 + 18) in 63 trials: 47 (1) to prevent collision and 23 (5 + 18) to recover from a path planning failure because of loop closure problems. Also, (2) Navigating to a different room revealed to be more difficult, with 72.8% success rate for autonomous navigation compared to 83.5% of (1) Navigating in a room. These difficulties can be explained by the following:
• The robot’s odometry influences navigation performance. SAM’s Odometry is calculated by the Beam base using wheel encoders and an inertial measurement unit. Rota-
Table 1: Navigation results from 10 home environments

| Number of trials | Trials successful autonomously (%) | Trials successful with intervention (%) | Interventions to prevent a collision (%) | Interventions because of path planning failure (%) | Interventions to help docking into the charging station (%) |
|------------------|-----------------------------------|----------------------------------------|----------------------------------------|-----------------------------------------------|---------------------------------------------------------|
| (1) Navigate in a room | 97 | 81 (83.5) | 94 (96.9) | 11 (9.3) | 5 (4.1) | 0 (0) |
| (2) Navigate to a different room | 173 | 126 (72.8) | 170 (98.3) | 36 (17.3) | 18 (8.7) | 0 (0) |
| (3) Return to the charging station located in the same room | 51 | 33 (64.7) | 39 (76.5) | 1 (2.0) | 3 (3.9) | 19 (27.5) |
| (4) Return to the charging station located in a different room | 79 | 45 (57.0) | 65 (82.3) | 16 (17.7) | 11 (11.4) | 25 (24.1) |
| **Total** | **400** | **288 (72.0)** | **368 (92.0)** | **64 (13.5)** | **37 (7.5)** | **44 (25.4)** |

When mapping, RTAB-Map memorizes images with their visual features as references for loop closure. For example, the top room of Environment B in Figure 17a is a laundry room. The first time the room was mapped, it had colorful clothes folded on the ironing table. The next day, RTAB-Map was unable to perform a loop closure because the clothes were gone and the colorful features were not visible. This problem can occur in all kinds of context in the homes like mapping dishes, food, shoes, clothes, pets, chairs, plants or even doors opened or closed. When RTAB-Map is unable to perform loop closure, the odometry error accumulates and the local map drifts from the global map. If the drift becomes too large, RTAB-Map is unable to find a possible path to satisfy both the local map and the global map, making autonomous navigation impossible. In this situation, the operator has to intervene and manually navigate the robot until RTAB-Map can perform a loop closure. Loop closure occurs when a match is found between the current image and an image in memory, using similarity measures based on visual features in the images.

When mapping, RTAB-Map recognizes the path element error of up to 10°, which decreases the quality of the map derived by RTAB-Map. While mapping and after finishing the mapping process, RTAB-Map can perform a loop closure. For each rotation in place, Odometry accumulates an error of up to 10°, which decreases the quality of the local map derived by RTAB-Map. For each rotation in place, Odometry accumulates an error of up to 10°, which decreases the quality of the local map derived by RTAB-Map. For each rotation in place, Odometry accumulates an error of up to 10°, which decreases the quality of the local map derived by RTAB-Map. For each rotation in place, Odometry accumulates an error of up to 10°, which decreases the quality of the local map derived by RTAB-Map. For each rotation in place, Odometry accumulates an error of up to 10°, which decreases the quality of the local map derived by RTAB-Map.
thus increasing the risk of collisions. When the robot had to get around these objects, the operator sometimes had to intervene to prevent a collision. Also, having set the minimum obstacle height at 20 cm led to a problem in detecting a walker in Environment D, as illustrated by Figure 15, and ignoring small objects on the floor like shoes. If the robot planned paths toward misinterpreted or ignored objects, the operator had to intervene to deviate the trajectory to avoid collision. When navigating to a different room, an increase of proportion of interventions happened to prevent collision (from 9.3% to 17.3%) or because of path planning failure (from 4.1% to 8.7%). This growth is caused by odometry drift when navigating through door frames. Door frames are narrow space that allow no room for error, and if the local map is not aligned with the global map, the robot can plan a path too close to the door frame or can be incapable to find a path. In these circumstances, the operator had to intervene. If we had only considered trials performed in senior residences (environments D, F and G), SAM autonomous success rate would have increased to 89% over 70 trials, which can be explained by the fact that tight spaces, narrow turns and furniture near door frames were almost nonexistent in these environments. This decreases the occurrences of having an obstacle in the robot’s blind spot and helps find a valid path despite a drift in the local map. Thus, large door frames adapted for wheelchair are more permissive for odometry drift, as observed in Table 2.

Regarding the autonomous navigation scenarios (3) and (4) of Table 1 involving the charging station, in addition to have to face navigation challenges outlined above, SAM experienced difficulties docking in some cases, requiring interventions 44 times over 25.4% of the trials involving this modality. As shown by Figure 8, depending on illumination conditions, the symbol on the flat part of the charging station may not be defined enough, generating orientation errors of up to 20°. Also, the flat part of the charging station is made of metal, which has low friction: if SAM’s propelling wheels move from a high friction surface (e.g., carpet, anti-slip lenoleum) to a low friction surface, the wheels sometimes spin for a short time because the motor controller temporarily over-shoots the amount of power sent to the motors. This makes the robot deviate from its planned trajectory, making it unable to dock correctly. Special care should be put on placing the charging station over a surface with low friction to facilitate docking.

### 6.2 Autonomous conversation following

Table 3 presents the results of autonomous conversation following done in environments A, B, E and J from Table 4. In quiet condition, SAM succeeded in directing the camera towards the person talking 93% of the time, and in the remaining 7% the robot remained still. In noisy conditions performance dropped to 62%. Interfering sound sources which included voices, such as television and music lyrics, were sometimes considered as a valid interlocutor, making the robot turn towards it. On the other hand, kitchen hood and vacuum cleaner noise were very rarely detected as voice. This is made possible by ODAS’ voice detection algorithm, which analyzes the frequency range of the sound source. The voice of a male adult has a fundamental frequency from 85–180 Hz and 165–255 Hz for female adult [63]. If the interfering sound overlaps in the 85–255 Hz interval, false recognition may occur.

| Scenario                  | Quiet | Noisy |
|---------------------------|-------|-------|
| Change of interlocutor    | 288   | 252   |
| Followed successfully     | 93%   | 62%   |

---

**Table 2: Autonomous navigation through door frames**

| Door width (cm) | Number of attempts | Successful autonomously (%) |
|-----------------|--------------------|----------------------------|
| 58              | 24                 | 79                         |
| 71              | 132                | 78                         |
| 76              | 85                 | 77                         |
| 83              | 46                 | 98                         |
| Total           | 287                | 81                         |

**Figure 15:** Object misinterpretation caused by the minimum height threshold. Light gray area shows ideal 2D representation of the obstacle and dark gray shows misinterpretation.
Table 4: Description of the 10 real homes used for the trials

| ID | Home type       | Floor type            | Door frame width (cm) | Hallway width (cm) | Door threshold height (cm) | Rooms                                                  | Furniture                                                                 |
|----|-----------------|-----------------------|-----------------------|--------------------|---------------------------|--------------------------------------------------------|---------------------------------------------------------------------------|
| A  | Apartment       | Hardwood              | 71                    |                    |                           | Bedroom and Living Room                                | Bed, nightstand, dressers, mirror, sofas, coffee tables and Christmas tree |
| B  | Basement        | Rough Carpet          | 76                    |                    |                           | Laundry Room and Living Room                           | Ironing table, counters, washer, dryer, desk, sofas, chair, table and filing cabinet |
| C  | Ground Floor    | Hardwood              | 71                    | —                  | 0.3                       | Living Room, Entrance Room, Bathroom, Dining Room and Kitchen Bedroom, Living Room, Bathroom, Kitchen and Hallway | Sofas, coffee table, TV, cat tree, counter, mirror, toilet, washer, dryer, table, chairs and counters |
| D  | Senior Residence| Laminate              | 76, 83                | 110                | 0.3                       | Living room, Bedrooms, Hallway                         | Bed, dresser, walker, armchairs, tables, TV, chairs, counters, bath, toilet and mirror |
| E  | Ground Floor    | Hardwood              | 71                    | 86                 | 0.4                       |                                                        | Sofas, rug, coffee table, plant, bed, dresser, nightstand and bed          |
| F  | Senior Residence| Anti-Slip Linoleum    | 83                    |                    |                           | Living Room, Bedroom, Kitchen and Bathroom             | Armchairs, TV, coffee table, bed, dresser, counter, table, chairs, counter, bath and toilet |
| G  | Senior Residence| Anti-Slip Linoleum    | 83                    |                    |                           | Living Room, Office, Kitchen, Dining Room and Bedroom  | Armchairs, TV, coffee table, desk, chairs, mirror, counter, table, bed and dresser |
| H  | Ground Floor    | Laminate              | 58, 71, 76            | 112                |                           | Living Room, Empty Room                                | Sofas, armchair and round table                                         |
| I  | Basement        | Rough Carpet          | 58, 71                | 112                |                           | Laundry Room, Walk-ins, Training room and Hallway      | Counter, ironing table, shelves and training equipment                  |
| J  | Apartment       | Laminate              | 76                    | 86                 | 0.3                       | Entrance Room, Bedroom and Hallway                    | Bed, nightstand, desk, chair, dresser and mirror                        |
Overall, this functionality could enhance remote experience and would be ready for conducting usability study in quiet conditions but would need to be improved for noisy conditions possibly by memorizing the signature of acceptable sounds to track [64].

7 Limitations of the work

The work presents in details the implementation of SAM, our telepresence mobile robot prototype designed for remote home care assistance. It reveals insights and issues when designing and integrating autonomous decision-making capabilities for a mobile robot platform and experimenting in real home settings. However, this article reports on what can be considered a first step in providing autonomy to robots for remote care assistance, and it is important to outline the following limitations of the work:

- Results presented are limited by SAM’s hardware and software components, which constrained our experimental methodology, and we had to adapt to limitations observed (as indicated in Section 5), making it more exploratory in nature. Results would also differ using a different robotic platform with other algorithms for the robot control architecture, autonomous navigation and sound processing. However, providing a detailed description of SAM’s implementation and observations made in the field can serve as a reference for future comparative studies. We are currently improving what was implemented with SAM and porting its implementation on other robot platforms with improved sensing (e.g., adding a laser, and IMU on top of the robot and a second RGB-D camera to cover the blind spot, monitor vibrations and better odometry for more robust navigation; doing mapping continuously adding semantic mapping [65] to remove less reliable visual features, visual SLAM in illumination changing conditions [66]), interaction (e.g., people finding using face recognition and sound source localization; filtering noise using sound source separation) and teleoperation (e.g., the use of 3D representations for navigation) capabilities. Developing these improved capabilities through HBBA, RTAB-Map and ODAS will facilitate prototyping while allowing others to exploit them on their own implementation.

- SAM is designed to be a research prototype and not a commercial product. Many steps would have to be taken to make it commercially ready and to comply with ISO 13482:2014 standard (Robots and robotic devices – Safety requirements for personal care robots). Any new design or integration to an existing platform would have to take these elements into consideration.

- Trials conducted in the 10 home environments in controlled supervision are not representative of realistic large-scale and long-term deployment conditions. Trials in a more diverse set of home environments with living occupants are required. In addition to making the autonomous capabilities of the robot more robust for such trials, we are currently developing our own cloud-based middleware to implement end-to-end telehealth solutions [67] to support such deployment.

- The experimental methodology followed does not involve usability and interaction studies with SAM in remote care assistance nor for the teleoperation with autonomous capabilities, as conducted for instance in the ExCITE project described in Section 2. The current work must be considered as a stepping stone toward such types of experiments. Our focus will be on senior residences, in which assistance for the robot is more likely to be available from onsite technical staff. Following a user-centered design approach involving clinicians, seniors and caregivers, our strategy is to first conduct demonstration trials to residents to illustrate what can be done with the robot, to help co-construct interaction scenarios to be conducted with SAM and other robotic platforms.

8 Conclusion

This article outlines the different elements that come into play to provide autonomous capabilities to a telepresence robot, from navigation to interaction modalities and their hardware, software and decision-making integration. It reveals to be quite a challenge to provide reliability and robustness of the different autonomous modalities integrated on SAM, and our experiments in real home settings identify the capabilities and also the limitations of the platform that were not experienced in lab conditions. In spite of those limitations, SAM performed reasonably well in real home settings, and we learned a lot from conducting trials in real homes, pointing out interesting issues to work on. This suggests that conducting trials evaluating the robot’s autonomous capabilities in real conditions would be more reliable and representative of real living conditions.
home settings is an important preliminary step because it makes it possible to outline what can be expected of the robot and derive interaction scenarios in accordance with the robot’s capabilities. Identifying these limitations prior to conducting usability studies, for instance, which require significant time and resources, makes it possible to characterize how autonomous capabilities will influence methodology and results the operating environments will have to be either constrained or engineered in some ways, the robot and its autonomous capabilities will have to be improved to make them more robust, or the limitations will have to be taken into consideration when analyzing the results. Robot platform will have to change to minimize the occurrences of those limitations. SAM’s autonomous capabilities are imperfect, and not acknowledging or understanding them could lead to invalid observations if not considered when planning and conducting usability studies.

In our future work, we will continue to strive for autonomy and trials in real home environments, which we believe are key to address minimal requirements for the safety and failure safe operation of the robot, the assistance of remote operator and can even be a solution for ethical issues of privacy [19]. We also hope that continuing to make our code available and facilitate accessibility to technologies will help forge new partnerships and collaborations working toward enhancing autonomy of telepresence mobile robots for remote care assistance.

Acknowledgements: This work was supported by AGEWELL, the Canadian Network of Centres of Excellence on Aging Gracefully across Environments using Technology to Support Wellness, Engagement, and Long Life.

Conflict of interest: François Michaud is an Editor of the Paladyn, Journal of Behavioral Robotics and was not involved in the review process of this article.

Data availability statement: Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

References

[1] M. White, M. V. Radomski, M. Finkelstein, D. A. S. Nilsson, and L. I. E. Oddsson, “Assistive/socially assistive robotic platform for therapy and recovery: Patient perspectives,” Int. J. Telemed. Appl., vol. 2013, p. 11, 2013.

[2] C. Mucchiani, W. O. Torres, D. Edgar, M. J. Johnson, P. Z. Caccione, and M. Yim, “Development and deployment of a mobile manipulator for assisting and entertaining elders living in supportive apartment living facilities,” in Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, 2018, pp. 121–128.

[3] Canada Health Infoway, “Telehealth Benefits and Adoption Connecting People and Providers Across Canada,” Canada: Praxia Information Intelligence, 2011.

[4] A. Orlandini, A. Kristoffersson, L. Almqvist, P. Björkman, A. Cesta, G. Cortellessa, et al., “ExCITE Project: A review of forty-two months of robotic telepresence technology evolution,” Presence, vol. 25, pp. 204–221, Dec 2016.

[5] J. M. Beer and L. Takayama, “Mobile remote presence systems for older adults: Acceptance, benefits, and concerns,” in Proceedings of the International Conference on Human-Robot Interaction, HRI ’11, 2011, pp. 19–26.

[6] I. Orha and S. Oniga, “Assistance and telepresence robots: A solution for elderly people,” Carpathian J. Electr. Eng., vol. 5, pp. 87–90, 2012.

[7] A. Reis, R. Xavier, I. Barroso, M. J. Monteiro, H. Paredes, and J. Barroso, “The usage of telepresence robots to support the elderly,” in Proceedings of the International Conference on Technology and Innovation in Sports, Health and Wellbeing, June 2018, pp. 1–6.

[8] S. Laniel, D. Létourneau, M. Labbé, F. Grondin, J. Polgar, and F. Michaud, “Adding navigation, artificial audition and vital sign monitoring capabilities to a telepresence mobile robot for remote home care applications,” in Proceedings of the International Conference on Rehabilitation Robotics, July 2017, pp. 809–811.

[9] T.-C. Tsai, Y.-L. Hsu, A.-I. Ma, T. King, and C.-H. Wu, “Developing a telepresence robot for interpersonal communication with the elderly in a home environment,” Telemed. e-Health, vol. 13, no. 4, pp. 407–424, 2007.

[10] T. S. Dahl and M. N. K. Boulou, “Robots in health and social care: A complementary technology to home care and telehealthcare?” Robotics, vol. 3, no. 1, pp. 1–21, 2013.

[11] A. Kristoffersson, S. Coradeschi, and A. Loutfi, “A review of mobile robotic telepresence,” Adv. Hum. Comput. Interact., vol. 2013, p. 3, 2013.

[12] T. Lewis, J. Drury, and B. Beltz, “Evaluating mobile remote presence (MRP) robots,” in Proceedings of the 18th International Conference on Supporting Group Work, ACM, 2014, pp. 302–305.

[13] M. Desai, K. M. Tsui, H. A. Yanco, and C. Uhlik, “Essential features of telepresence robots,” in Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications, 2011, pp. 15–20.

[14] K. M. Tsui, M. Desai, H. Yanco, and C. Uhlik, “Exploring use cases for telepresence robots,” in Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, 2011, pp. 11–18.

[15] K. M. Tsui and H. A. Yanco, “Design challenges and guidelines for social interaction using mobile telepresence robots.”, Rev. Hum. Factors Ergon., vol. 9, no. 1, pp. 227–301, 2013.

[16] M. K. Lee and T. Takayama, “Now, I have a body”: Uses and social norms for mobile remote presence in the workplace,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2011, pp. 33–42.

[17] I. Mendez, M. Jong, D. Keays-White, and G. Turner, “The use of remote presence for health care delivery in a northern Inuit
[47] F. Ferland, R. Agrigoroaie, and A. Tapus, “Assistive humanoid robots for the elderly with mild cognitive impairment,” in Humanoid Robotics: A Reference, A. Goswami and P. Vadakkepat, Eds., Springer Netherlands, 2018.

[48] C. Stachniss, Robotic Mapping and Exploration, vol. 55, Springer-Verlag Berlin Heidelberg: Springer Science & Business Media, 2009.

[49] M. Labbé and F. Michaud, “Appearance-based loop closure detection for online large-scale and long-term operation,” IEEE Trans. Robot., vol. 29, pp. 734–745, June 2013.

[50] M. Labbé and F. Michaud, “Long-term online multi-session graph-based SLAM with memory management,” Autonom. Robots, vol. 42, pp. 1133–1150, 2018, DOI: https://doi.org/10.1007/s10514-017-9682-5.

[51] M. Labbé and F. Michaud, “RTAB-Map as an open-source lidar and visual SLAM library for large-scale and long-term online operation,” J. Field Robot., vol. 36, no. 2, pp. 416–446, 2018.

[52] T. Moore and D. Stouch, “A generalized extended Kalman filter implementation for the Robot Operating System,” in Proceedings of the 13th International Conference on Intelligent Autonomous Systems, Cham: Springer, July 2014.

[53] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard, “Efficient estimation of accurate maximum likelihood maps in 3D,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2007, pp. 3472–3478.

[54] R. Wolff, M. Lasseck, M. Hild, O. Vilarroya, and T. Hadzibeganovic, Towards human-like production and binaural localization of speech sounds in humanoid robots, in Proceedings of the International Conference on Bioinformatics and Biomedical Engineering, 2009, pp. 1–4.

[55] F. Grondin and F. Michaud, “Lightweight and optimized sound source localization and tracking methods for open and closed microphone array configurations,” Rob. Autonom. Syst., vol. 113, pp. 63–80, 2019.

[56] A. Reveleau, F. Ferland, and F. Michaud, “Visual representation of interaction force and sound source in a teleoperation user interface for a mobile robot,” J. Hum. Robot. Interact., vol. 4, no. 2, pp. 1–23, 2015.

[57] F. Grondin, D. Létourneau, F. Ferland, V. Rousseau, and F. Michaud, “The Manyeyes open framework,” Autonom. Robots, vol. 34, no. 3, pp. 217–232, 2013.

[58] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, et al., “ROS: An open-source Robot Operating System,” in ICRA Workshop on Open Source Software, vol. 3, pp. 5–11, 2009.

[59] E. Ackerman, “iRobot and InTouch Health announce RP-VITA telemedicine robot,” IEEE Spectrum, July 2012.

[60] J. Gonzalez-Jimenez, C. Galindo, and J. R. Ruiz-Sarmiento, “Technical improvements of the Giraff telepresence robot based on users’ evaluation,” in Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, 2012, pp. 827–832.

[61] P. Lepage, D. Létourneau, M. Hamel, S. Brière, H. Corriveau, M. Tousignant, and F. Michaud, “Telehomecare telecommunication framework — from remote patient monitoring to video visits and robot telepresence,” in Proceedings of the IEEE International Conference on Engineering in Medicine and Biology Society, Aug 2016, pp. 3269–3272.

[62] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 7, pp. 1409–1422, 2011.

[63] J. F. Mahdi, “Frequency analyses of human voice using fast Fourier Transform,” Iraqi J. Phys., vol. 13, no. 27, pp. 174–181, 2015.

[64] F. Grondin and F. Michaud, “WISS, a speaker identification system for mobile robots,” in Proceedings of the IEEE International Conference on Robotics and Automation, 2012, pp. 1817–1822.

[65] J. Vincent, “Masquage sémantique d’instances pour SLAM visuel dans des environnements dynamiques,” Master’s thesis, Departement of Electrical Engineering and Computer Engineering, Université de Sherbrooke, 2019.

[66] M. Labbé and F. Michaud, “Multi-session visual SLAM for illumination invariant localization in indoor environments,” in Proceedings of the IEEE International Conference on Robotics and Automation (submitted), 2021.

[67] A. Panchea, D. Létourneau, S. Brière, M. Hamel, M.-A. Maheux, C. Godin, et al., “OpenTera: A microservice architecture solution for rapid prototyping of robotic solutions to COVID-19 challenges in care facilities,” Front. AI Robot. (in revision), 2021.
Appendix A  Description of the 10 homes used for the trials

Figure 16: Environment A. (a) Sketch (b) RTAB-MAP.

Figure 17: Environment B. (a) Sketch (b) RTAB-MAP.
Figure 18: Environment C. (a) Sketch (b) RTAB-MAP.

Figure 19: Environment D. (a) Sketch (b) RTAB-MAP.
Figure 20: Environment E. (a) Sketch (b) RTAB-MAP.

Figure 21: Environment F. (a) Sketch (b) RTAB-MAP.
Figure 22: Environment G. (a) Sketch (b) RTAB-MAP.

Figure 23: Environment H. (a) Sketch (b) RTAB-MAP.
**Figure 24:** Environment I. (a) Sketch (b) RTAB-MAP.

**Figure 25:** Environment J. (a) Sketch (b) RTAB-MAP.