Attempting to Have Mobile Robots Move to Real Homes Through the Development of Open Source Libraries

François Michaud, Mathieu M. Labbé, Dominic Létourneau,
François Ferland, Sébastien Laniel
Interdisciplinary Institute for Technological Innovation (3IT)
Université de Sherbrooke
Sherbrooke, Québec, Canada
{given name.surname}@usherbrooke.ca

François Grondin
CSAIL Laboratory
Massachusetts Institute of Technology
Cambridge, USA
fgrondin@mit.edu

***Abstract*—One of the greatest demographic and social transformations facing developed countries is caused by the aging of their populations, increased life expectancy and related challenges. Incidentally, this supports the fast growing development of technology used to provide home care, including robotics. In particular, mobile telepresence robotic platforms are now commercially available and provide mobility to sensors, actuators and interactive devices into real world settings, without having to engineer the environment for their use. However, usability of these platforms for such applications requires that they be equipped with some autonomy for navigation and interaction. This paper presents three open source libraries developed to address the challenges of navigation, artificial audition and integration that we have been developing for the design of a home assistance robot. These libraries are being developed with real-time, limited processing and robustness requirements in mind so that they work out of the lab and into real homes. The current usage of these libraries is illustrated using SAM, an enhanced Beam+ platform.**

***Index Terms*—Open source libraries, Simultaneous Planning, Localization and Mapping, Artificial audition, Robot control architecture**

I. Introduction

Our interest lies in the design of service and assistive robots for personal domestic applications, and more specifically a mobile robot to conduct virtual visits for remote consultations or assistance, in a socially and economically responsible fashion. Even though mobile telepresence robotic platforms have recently been introduced on the market [1] [2], what is missing for making them effective remote home care assistance systems are capabilities specifically needed to assist the remote operator, whom would most likely be novice robot users (e.g., clinicians, caregivers), in conducting such virtual visits, and to avoid having the occupant provide assistance to the robot. Basic specifications envisioned are:

- **Autonomous recharging**: The robot must be able to recharge autonomously by navigating to its charging station. This is an essential feature to avoid having the robot be teleoperated back to its charging station by the remote operator at the end of a session (to save time), or to be moved by the occupant in case of a telecommunication failure or low energy level. This requires the robot to have mapping and localization capabilities, allowing it to navigate efficiently in the home.
- **Efficient interaction with people**: To minimize cognitive load and maximize situation awareness [3], a remote operator would find it beneficial to receive assistance in following and tracking people which whom to interact. Such capabilities would minimize what the remote operators have to do to control the platform while still perceive the appropriate information to focus on the interaction tasks to be conducted through telepresence.
- **Expandability and portability**: With continuous technological progress and the availability of mobile robot platforms having higher processing and interacting capabilities, it is important to use an integration framework allowing to expand and port the system in accordance with user's needs and the sensing, reasoning and acting capabilities of the robot platform.

This paper presents three open source libraries that we have been developing to address such specifications: RTAB-Map [4], [5], ODAS [6] and HBBA [7], [8]. These libraries are developed by always taking into consideration the constraints of having to operate in real world settings using robots that have limited processing capabilities. We illustrate their use in the design of SAM [9], our remote assistance robot platform designed on a Beam+ platform from Suitable Technologies.

II. RTAB-Map

SPLAM (Simultaneous Planning, Localization And Mapping) [10] 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 on a robot with limited computing resources. A key feature in SPLAM is detecting previously visited areas to reduce map

This work is supported by AGE-WELL, the Network of Centres of Excellence on Aging Gracefully across Environments using Technology to Support Wellness, Engagement, and Long Life.

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 the computing power required is 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 detections are 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)[1] [4], [5] is our open source library implementing such an approach, using images of the operating environment. Released in 2013, RTAB-Map can be used as a cross-platform standalone C++ library and with its ROS package[2] to do 2D or 3D SLAM. The standalone binaries have been downloaded more than 13000 times and its GitHub repository has around 20 clones and 600 visits per day. RTAB-Map has been also used by the winners of the Microsoft Kinect Challenge held at IROS 2014[3].

Figure 1 illustrates an example of a 3D and a 2D map representations created with RTAB-Map using a Kinect camera and a 2D lidar. 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 [11]. 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 [12]. 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, 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 determined 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 detections and to extend the current local map.

[1] http://introlab.github.io/rtabmap

[2] http://wiki.ros.org/rtabmap_ros

[3] goo.gl/UJiPzZ

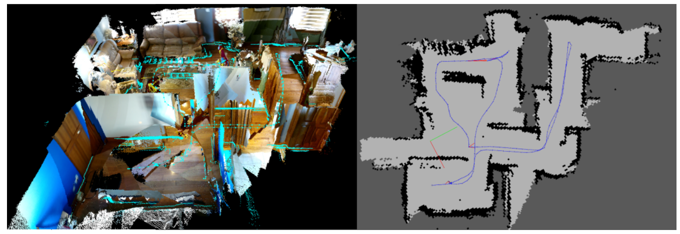


Fig. 1: Map generated by RTAB-Map

III. ODAS

Similarly to artificial vision, artificial audition consists of improving the ability to derive auditory information about the remote environment from the robot platform. Robots for home assistance would operate in noisy environments, and limitations are observed in such conditions when using only one or two microphones [13]. Using a microphone array can enhance performance by allowing a robot to localize, track, and separate multiple sound sources to improve situation awareness. ODAS [14] is our newest open source library[4] performing sound sources localization, tracking and separation. ODAS is derived from our previous work on ManyEars [6], also distributed as an open source library[5]. ManyEars uses the generalized cross-correlation phase transform for audio localization, and a particle filter based method for tracking. Released in 2009, ManyEars has been downloaded more than 8355 times. ManyEars was used in December 2009 by the winning team *Fly By Ear* of the Annual Machine Intelligence Competition, run by the British Computer Society.

Figure 2 shows the main components inside the ODAS framework. ODAS improves robustness to noise by allowing to increase the number of microphones used while reducing computational load. This library relies on a new 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 novel 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 [15]. 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. This new library also models microphones as sensors with a directive polar pattern, which improves sound sources localization, tracking

[4] http://odas.io

[5] https://github.com/introlab/manyears

and separation when the direct path between microphones and the sound sources is obstructed by the robot's body. Figure 3 presents ODAS Studio application[6] displaying real time 3D sound source locations for fine tuning of microphone positions and ODAS parameters.

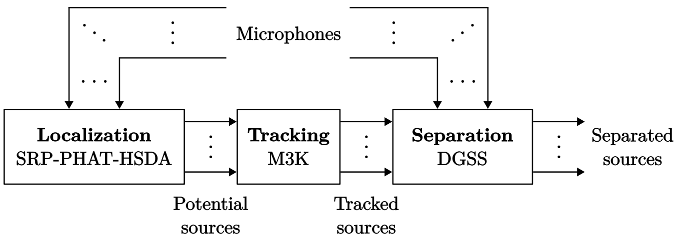


Fig. 2: ODAS architecture

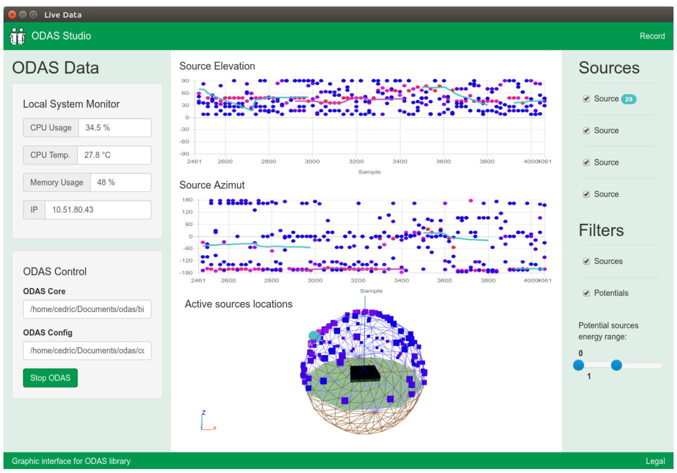


Fig. 3: ODAS studio application

To make use of ODAS or ManyEars, 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 such as sound effects, integrated mixing, optical inputs/outputs, S/PDIF, MIDI, numerous analogs outputs, etc., which are not required for artificial audition; they also require significant amount of power and size. To facilitate the use of ODAS and ManyEars on various robotic platforms, we also provide as open hardware two sound cards: 8SoundsUSB[7] and 16SoundsUSB[8] (released in January 2018), for eight and sixteen microphone arrays, respectively. They provide synchronous acquisition of microphone signals through USB to the robot's computer. The 8SoundsUSB kit has been downloaded more than 8850 times, and is used by research labs around the world [16]–[18].

IV. HBBA

Design of an interactive mobile robot is one [19] if not the most challenging integration problem in robotics. It involves dealing with action (manipulation, mobility), perception (environment, people), interaction (information exchange modalities such as interpretation of perceptual cues, human-robot interfaces, etc.), systems (mechatronics, control, software, cognition) [20], in relation to an application domain. These elements are all interdependent, as each one influences the others.

To address this challenge, robot control architectures define the interrelations between decision-making modules required the application. There is an infinite number of ways to implement robot control architectures, and 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 and human-robot interaction (HRI) [21]. Integration and coordination of different types of processing modules (perception, reasoning, behaviors) is one significant challenge, and there is currently no dominant solution [22].

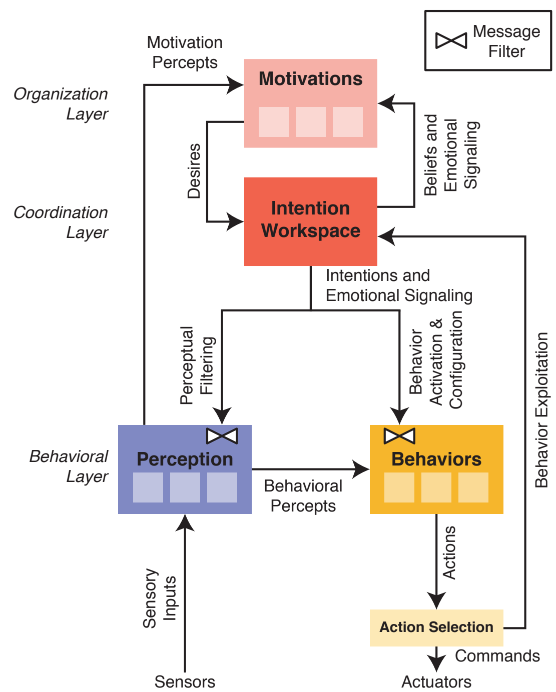


Fig. 4: Hybrid Behavior-Based Architecture (HBBA)

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 [23]. For instance, the Donaxi robot [24], [25] has a deliberative (for symbolic representation and reasoning), an executive (for plan monitoring) and a functional layer. Siepmann et al. [26] uses a hardware, a functional and a BonSAI layer. The complexity in layered robot control architecture comes in how to interface and partition these layers [27]. To address these issues, we are designing HBBA (Hybrid Behavior-Based Architecture) [7], [8] as an open

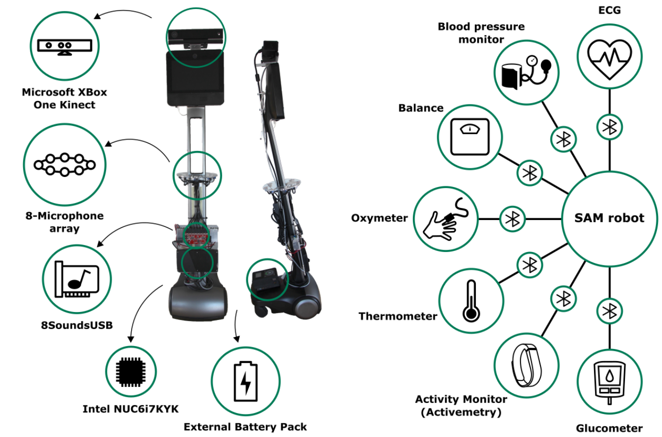
[6]https://github.com/introlab/odas_web
[7]https://sourceforge.net/projects/8soundsusb/
[8]https://github.com/introlab/16SoundsUSB

source[9] and unifying framework for integrated design of human-robot interaction scenarios. Illustrated by Fig. 4, HBBA is a behavior-based architecture with no central representation that provides the possibility high-level modeling, reasoning and planning capabilities through Motivations 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, Behavior and Motivation modules 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 allows 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 [28], 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 behavior-producing modules has demonstrated on multiple robotic platforms 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. For instance, it has been used with humanoid robots such as the NAO and Meka Robotics M1 in a episodic memory sharing setup [29], and with the Robosoft Kompai and later on the PAL Robotics TIAGo as service robots for the elderly with mild cognitive impairments [30].

V. SAM, our Remote Assistance Robot

For the robot platform, we chose to enhance a Suitable Techologies Beam platform because of its low cost (2,140 \$US), its payload capability, and the possibility of interfacing it using a library presented in [31]. It 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 two hours of autonomy and a charging station. Beam's dimensions are 54.9" (H) × 12.3" (L) × 16.4" (D). Motor control and power management is accomplished via an USB 2.0 controller in the robot's base and its maximum speed is 0.45 m/s. Shown by Figure 5a, to interface our open source libraries, we placed a XBOX One Kinect camera on top of the LCD screen using custom made aluminium brackets. We installed a circular microphone array using the 8SoundsUSB [6] sound card using custom made aluminum brackets and acrylic support plates. For processing power, 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 original head computer's hard drive with a 128 GB mSATA drive. Both computers run Ubuntu 16.04 operating system with ROS (Robot Operating Systems [32]) Kinetic. A low-cost USB dongle is installed on the robot to acquire vital signs from battery-powered Bluetooth Low Energy (BLE) sensors shown in Figure 5b [33]. To separate the added components and the original robot, we added a SWX HyperCore 98Wh V-Mount lithium-ion battery on the robot's base using a V-Mount battery plate (to keep the robot's center of gravity as low as possible and to facilitate battery swapping and charging). This allow 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. Overall, the additional components cost a total of 2,300 \$US.

[9] http://github.com/francoisferland/hbba



(a) Added Hardware Components (b) BLE sensors

Fig. 5: SAM, our robot platform for home care assistance

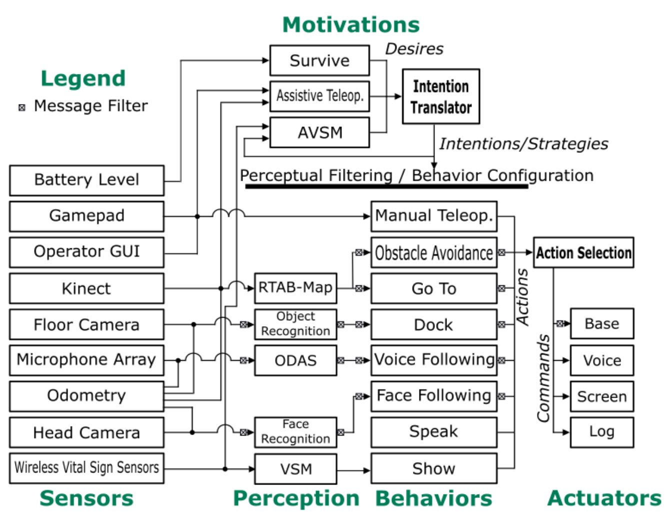


Fig. 6: SAM's robot control architecture using HBBA framework

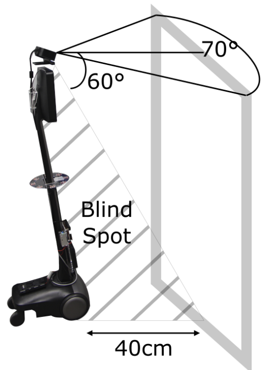


Fig. 7: SAM's field of view

Figure 6 illustrates what is implemented on SAM following the HBBA framework. The three main motivations are Survive, Assistive Teleoperation and Autonomous Vital Sign Monitoring. Survive supervises the battery level and generates a Desire to go to the charge station when battery level is too low. Using the interface, the remote operator can activates autonomous functionalities managed by the Assistive Teleoperation. This allows the user to either manually control the robot, to communicate high level destination for autonomous navigation, to follow autonomously a conversation or to autonomously orientate towards a person. AVSM (Autonomous Vital Sign Monitoring) allows the robot to use the same modalities in a sequence defined by a finite state machine. The scenario makes the robot autonomously find a specific person, provide vocal assistance when taking vital sign, and return to its charge station. To do so, SAM navigates to predefined location in the home with RTAB-MAP. Using voice detection with ODAS, the robot stops and orientates towards the sound source. SAM then uses face recognition to identify the interlocutor and if there is a match, the robot asks the interlocutor to measure vital signs using the BLE sensors shown in Figure 5b. When the measures are taken, the robot navigates and docks autonomously to its charge station.

VI. Results

SAM robot has been tested in 10 real home environments : 3 senior residences, 2 basements, 3 first floors and 2 apartments. Preliminary results suggest that SAM achieved a success rate of 92% over 400 navigation commands on a total of 35 different maps. The operator had to intervene 81 times, including 68 to avoid a collision. Therefore, the robot completed 286 (72%) autonomous navigation tasks. The efficiency highly depends on the robot limitations and the complexity of the environment.

For instance, to map its environment, RTAB-Map uses both the Kinect cloud point and SAM's odometry calculated using wheel encoders (60 ticks per rotation) and an IMU (Inertial Measurement Unit). SAM's odometry can lead to angular errors of up to 10° following a rotation in place, but during translation odometry errors are small. This makes the robot's navigation and mapping better in rectangular areas like hallways compared to circular environments. Also, since the Kinect camera is also used for obstacle avoidance, it is placed on the top of the robot, aligned toward the ground. As shown by Figure 7, SAM has a 40 cm blind spot because of the Kinect's limited FOV (Field Of View). In tight space, the Obstacle Avoidance behavior may not detect small objects, making navigation difficult near tables, sofas, plants and bed. Navigating in wide area is easier than going through hallways and door frames. Table I shows the results of 287 navigation through door frame. Overall, SAM had a 81% success rate. Senior residences with 82 cm width door frame adapted for wheelchair shows the best results with a score of 98%.

TABLE I: Results of autonomous navigation through door frame

Door Width	58 cm	71 cm	76 cm	82 cm	Total
Number of Tests	24	132	85	56	287
Succeeded Autonomously	19	103	65	45	232
	79%	78%	77%	98%	81%

TABLE II: Results for conversation following

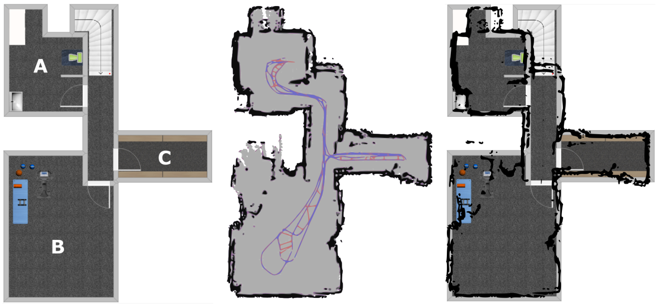
Environment	Quiet		Noisy	
Main Interlocutor Change	288		252	
Followed Main Interlocutor	267	93%	156	62%

Figure 8a illustrates the complexity of a real home. The wide open room B at the bottom was easy for the navigation algorithm but was hard to map because the cracked floor and drain caused wobble that led to misinterpreted images. The store room C at the right is a long narrow walk-in closet with a 58 cm door frame. The hallway is 112 cm and the room A at the top is a laundry room. Figure 8b presents one of the map generated by RTAB-Map, and Fig. 8c shows both the sketch and the overlapped map. Mapping a laundry room is a big issue because clothes are colorful and are in contrast to the background, making the mapping algorithm use these features as localization references. In localization mode, if these features are not found, RTAB-Map cannot correct its position. This problem occurs also with semi-dynamic objects like the different door angles and chair positions. To handle these issues, a classifier should be used to differentiate static (walls, frames, chandelier, etc.) and dynamic (bags, books, food, etc.) obstacles/features. Also, lighting must always be good for ensuring localization.

Table II shows the results of Conversation Following using the Voice Following Behavior. In a quiet environment (i.e., no loud interference were heard during the test), the robot was able to track an interlocutor 93% of the time. In a noisy scenario (e.g., sound of a kitchen hood, fan or music), performance drops to 62%.

VII. Conclusion

Designing a mobile robot for remote home care applications requires to keep in mind that the system must be easy to deploy, inexpensive and capable to adapt to various environments and operating conditions, and also be able to scale up by being able to follow technological progress. The three open source libraries presented in this paper have been designed to address



(a) Sketch (b) RTAB-MAP (c) Overlap

Fig. 8: Example of mapping and localization in a basement

real-time and limited processing capabilities of mobile robots, and to extend the capabilities of the platform when required. We have conducted and are in the process of analyzing the results a series of trials in different home environments to characterize SAM's performance using these libraries, before validating its actual use in senior residences.

REFERENCES

[1] A. Kristoffersson, S. Coradeschi, and A. Loutfi, "A review of mobile robotic telepresence," *Advances in Human-Computer Interaction*, vol. 2013, p. 3, 2013.

[2] T. Lewis, J. Drury, and B. Beltz, "Evaluating mobile remote presence (MRP) robots," in *Proc. 18th Int. Conf Supporting Group Work*, pp. 302–305, ACM, 2014.

[3] H. A. Yanco, J. L. Drury, and J. Scholtz, "Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition," *Human-Computer Interaction*, vol. 19, no. 1-2, pp. 117–149, 2004.

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

[5] M. Labbé and F. Michaud, "Long-term online multi-session graph-based SPLAM with memory management," *Autonomous Robots*, pp. 1 – 18, 2017.

[6] F. Grondin, D. Létourneau, F. Ferland, V. Rousseau, and F. Michaud, "The Manyears open framework," *Autonomous Robots*, vol. 34, no. 3, pp. 217–232, 2013.

[7] F. Ferland and F. Michaud, "Perceptual filtering for selective attention in a behaviour-based robot control architecture," *IEEE Trans. Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 256 – 270, 2016.

[8] F. Ferland, A. Reveleau, F. Leconte, D. Létourneau, and F. Michaud, "Coordination mechanism for integrated design of human-robot interaction scenarios," *Paladyn, Journal of Behavioral Robotics*, vol. 8, no. 1, 2017.

[9] 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 *Proc. International Conference on Rehabilitation Robotics*, pp. 809–811, July 2017.

[10] C. Stachniss, *Robotic Mapping and Exploration*, vol. 55. Springer Science & Business Media, 2009.

[11] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the Robot Operating System," in *Proc. 13th International Conference on Intelligent Autonomous Systems*, Springer, July 2014.

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

[13] 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 *Proc. Int. Conf. Bioinformatics and Biomedical Engineering*, pp. 1–4, 2009.

[14] F. Grondin and F. Michaud, "Lightweight and optimized sound source localization and tracking methods for open and closed microphone array configurations," *Robotics and Autonomous Systems*, pp. 63–80, 2019.

[15] 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. Human-Robot Interaction*, vol. 4, no. 2, pp. 1–23, 2015.

[16] A. Magassouba, N. Bertin, and F. Chaumette, "Audio-based robot control from interchannel level difference and absolute sound energy," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1992–1999, Oct 2016.

[17] C. Rascon and I. Meza, "Localization of sound sources in robotics: A review," *Robotics and Autonomous Systems*, vol. 96, pp. 184–210, 2017.

[18] V. Lunati, J. Manhès, and P. Danès, "A versatile System-on-a-Programmable-Chip for array processing and binaural robot audition," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 998–1003, Oct 2012.

[19] M. A. Goodrich and A. C. Schultz, "Human-Robot Interaction: A survey," *Foundation and Trends in Human-Computer Interaction*, vol. 1, pp. 203–275, http://dx.doi.org/10.1561/1100000005, Jan. 2007.

[20] T. W. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, pp. 143–166, 2003.

[21] "A Roadmap for U.S. Robotics – From Internet to Robotics," tech. rep., Robotics in the United States of America, Robotics VO, March 2013.

[22] SPARC, "Robotics 2020 Multi-Annual Roadmap, for Robotics in Europe, Release B." SPARC, The Partnership for Robotics in Europe (2015), Robotics 2020 Multi-Annual Roadmap, For Robotics in Europe, Release B, 2015.

[23] G. N. Saridis, "Intelligent robotic control," *IEEE Transactions on Automatic Control*, vol. AC-28, no. 5, pp. 547–557, http://dx.doi.org/10.1109/TAC.1983.1103278, 1983.

[24] H. S. Vargas, E. Olmedo, A. D. Martinez, V. Poisot, A. Perroni, A. Rodriguez, O. Granillo, C. Merino, C. Peto, W. Antonio, *et al.*, "Project donaxi home service robot," *Trans Tech Publications 2013*, vol. 423, pp. 2817–2820, 2013.

[25] H. S. Vargas, E. Olmedo, D. Martínez, V. Poisot, A. Perroni, A. Rodriguez, O. Granillo, C. Merino, W. Antonio, C. Carbonel, *et al.*, "Donaxi home project," *Team Description RoboCup@Home 2013*, 2013.

[26] F. Siepmann, L. Ziegler, M. Kortkamp, and S. Wachsmuth, "Deploying a modeling framework for reusable robot behavior to enable informed strategies for domestic service robots," *Robotics and Autonomous Systems*, vol. 62, no. 5, pp. 619–631, 2014.

[27] R. C. Arkin, *Behavior-Based Robotics*. MIT Press, 1998.

[28] G. D. Konidaris and G. M. Hayes, "An architecture for behavior-based reinforcement learning," *Adaptive Behavior*, vol. 13, no. 1, pp. 5–32, 2005.

[29] F. Ferland, A. Cruz-Maya, and A. Tapus, "Adapting an hybrid behavior-based architecture with episodic memory to different humanoid robots," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication*, pp. 797–802, 2015.

[30] 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.

[31] L. Xiang, "Mapping and control with telepresence robots," tech. rep., Department of Computer Science, Brown University, 2015.

[32] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, vol. 3, pp. 5–11, 2009.

[33] 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 *Proc. Annual Int. Conf. IEEE Engineering in Medicine and Biology Society*, pp. 3269–3272, Aug 2016.