A Cloud Robotics System for Telepresence enabling Mobility Impaired People to Enjoy the whole Museum Experience

Miguel Kaouk Ng*, Stefano Primatesta*, Luca Giuliano*, Maria Luce Lupetti*, Ludovico Orlando Russo*, Giuseppe Airò Farulla*, Marco Indaco*† Stefano Rosa*, Claudio Germak*, Basilio Bona*
*Politecnico di Torino
Corso Duca degli Abruzzi 24
I-10129 Torino TO Italy
E-mail: {name.surname}@polito.it
† Lero - The Irish Software Engineering Research Centre
Limerick, Ireland

***Abstract*—We present a novel robotic telepresence platform composed by a semi-autonomous mobile robot based on a cloud robotics framework, which has been developed with the aim of enabling mobility impaired people to enjoy museums and archaeological sites that would be otherwise inaccessible. Such places, in fact, very often are not equipped to provide access for mobility impaired people, in particular because these aids require dedicated infrastructures that may not fit within the environment and large investments. For this reason, people affected by mobility impairments are often unable to enjoy a part or even the entire museum experience. Solutions allowing mobility impaired people to enjoy museum experience are often based on recorded tours, thus they do not allow active participation of the user. On the contrary, the presented platform is intended to allow users to enjoy completely the museum round. A robot equipped with a camera is placed within the museum and users can control it in order to follow predefined tours or freely explore the museum. Our solution ensures that users see exactly what the robot is seing in real-time. The cloud robotics platform controls both navigation capabilities and teleoperation. Navigation tasks are intended to let the robot reliably follow pre-defined tours, while main concern of teleoperation tasks is to ensure robot safety (e.g., by means of dynamic obstacle detection and avoidance software). Proposed platform has been optimized to maximize user experience.**

I. INTRODUCTION

Mobile robotics platforms [1] are nowadays increasingly spreading into various fields of our daily life. This is mainly due to technology developments in the field of service robotics. Due to our cultural background we always tend to link robotics applications with the industrial production only. Present-day life reality, instead, consists of a world that is slowly getting populated by highly intelligent technological systems. Service robotics has been widely applied in assistive technologies [2], [3], rehabilitation [4], [5], [6], assistive mobility [7], remote communication systems for disabled people [8] and telepresence [9], [10], [11].

It seems that this new trend in robotics fights again human preconceptions. There exist a feeling of distrust against service and smart robotics, that has deeply settled historical reasons, basically deriving from negative examples of human-robot coexistence and failed human-robot interaction (HRI) paradigms. However, from a social point of view, humans are slowly getting used to share physical spaces with robotic platforms and the increasing presence of technological devices in cultural environments (thus outside laboratories and factories) is the proof of it. Taking into account this scenario, the research area about the interaction among robotics and cultural heritage assumes a very high importance and is today gaining momentum. The attempt to introduce robotic systems into museums has always fascinated researchers. Recently it is becoming possible to proof early systems where service robotic applications are employed to offer an added value to the museum experience and overcome limitations of current technologies used. Technologies applied to cultural heritage can be grouped into three areas [12]:

- **Audio guides**: typically a pair of headphones connected to a digital audio player, which provides a recorded spoken commentary to deepen the artworks descriptions. Visitors who use audio guides usually spend more time in front of the artworks, develop a stronger interest in the artist, and are more likely to report a positive emotional response [13].
- **Multimedia guides**: typically small-screens which provide audio-visual information that supporting the visit of the tourist.
- **Personal technologies**: portable media players such as smartphones that are used to visualize audio-visual clips and other additional information.

All these technologies, however, are meaningful only when the cultural site is accessible for a given user. Instead, mobile robotic technologies can also be applied to let people experience closed or inaccessible archaeological sites, and to assist mobility impaired people, to let them enjoy completely the museum experience, even if the museum is not providing dedicated infrastructure to fit their needs. In particular, resorting to telepresence systems, a remote user can interact a robot through a simple and intuitive web interface, while the robot itself is within the museum site, and the user can see exactly what the robot sees.

978-1-4799-1999-4/15/$31.00 ©2015 IEEE

Aim of this paper is to present and describe a service robotic application that uses a mobile robot as its core to enhance the museum experience. Our solution is based on the concept of *Cloud Robotics* [14], [15], that is a novel paradigm in robotics which aims at moving computational and storage resources from the robot to the cloud, in order to develop simpler and cheaper robots.

We have developed a custom cloud robotics platform based on the Robot Operating System (ROS) [16], that is an open-source, meta-operating system for robot software development, providing a collection of packages, software building tools and an architecture for distributed inter-process and inter-machine communication. ROS-based applications building blocks are the so-called nodes, i.e., pieces of code implementing a specific functionality. Nodes interact with each other relying on request/reply and publish/subscribe communication models. The communication between nodes is based on the TCP network protocol.

The rest of the paper is so organized: Section II presents the technological robotic solution we have developed; Section III introduces the proposed application; Section IV presents preliminary platform validation and finally Section V concludes the paper and discusses future works.

II. The Developed System

The proposed solution is based on a Cloud Robotics Platform (CRP, depicted in Fig. 1), that is a framework making use of huge resources available on the cloud to develop a more powerful and centralized intelligence for robotics application. The main idea within this framework is to convert the robot into a *cloud agent*, while most of the computation is done on remote servers. Moreover, the cloud robotics platform is also able to guarantee the robustness needed for long-term operativeness in the robotics applications and to expose simple APIs to the final user [17], [18], [19]. A schematic description of the framework used can be see in Figure. The mobile robot communicates with the CRP through a fast wireless internet connection, e.g., resorting on a Long Term Evolution (LTE) cellular data network. The CRP has a central unit called *Robot Clone Manager* (RCM) which manages all the resources needed to control the robot and to give feedback to the user. Basically, RCM manages and processes information from the robot sensors and exposes a web based Graphical User Interface (GUI) to the user.

RCM is also supported by a Platform Manager (PM) and a platform API manager (PAM). PM is the element in charge of managing data inside RCM. It implements an event-based logic, and is able to react to system notifications. PAM instead exposes simple RESTful APIs that can be used to interact with cloud servers.

The remaining of this Section aims at describing the proposed robotics architecture. In particular, Paragraph II-A illustrates the hardware architecture of the robotic platform used, Paragraph II-B presents the Cloud Robotic Platform architecture and Paragraph II-C illustrates the main features of the Web User Interface.

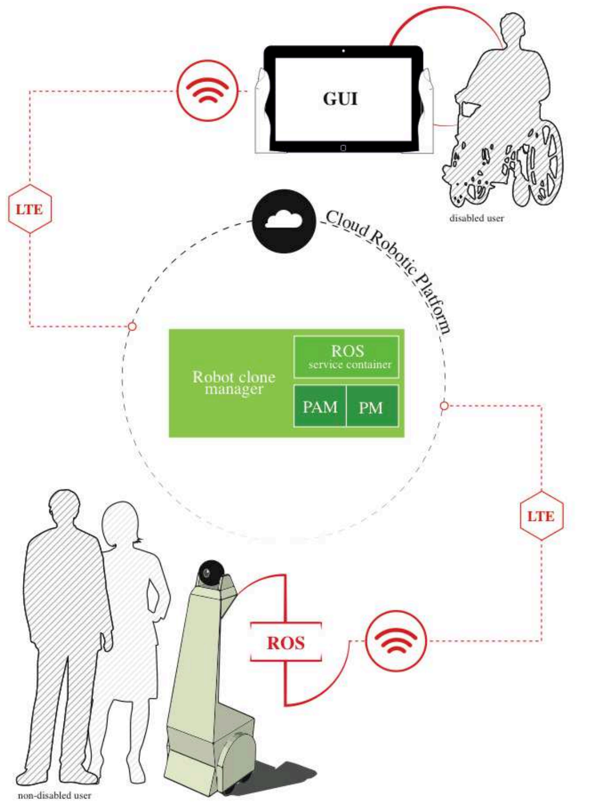

Fig. 1: Schematic description of the HRI paradigm. Users can remotely control the robotic platform through a simple web app and a smart device. Interaction with the robot is managed by the CRP framework.

A. Hardware Architecture

The mobile robot used is a four wheels drives capable of moving in indoor structured environments, specifically designed for purpose reasons. This robot implements a fully ROS compatible platform suitable for a wide range of activities. Its mechanical structure is designed in a pyramid shape with a rectangular base of 50 x 55 cm, a height of 120 cm and a weight of 10 kg approx. Wheels are operated by two electric motors, a set of gears and belts. Each motor provides traction to each side in a separate way. Robot autonomy is up to 8 hours; power is provided by a 12V Li-Fe battery and the robot is equipped with both proprioceptive and exteroceptive sensors to estimate its own motion. Its maximum speed is of 1 m/s and it can move in teleoperation mode through the assisted guidance of an user connected remotely via a web server application. It mounts several on-board sensors, including wheel encoders, a Hokuyo UTM-30LX laser range finder and a DCS-5222L Pan/Tilt camera.

The overall robot hardware architecture is depicted in Fig. 2. The Electronic Control Unit (ECU) manages the embedded applications, with particular regards to motor motion

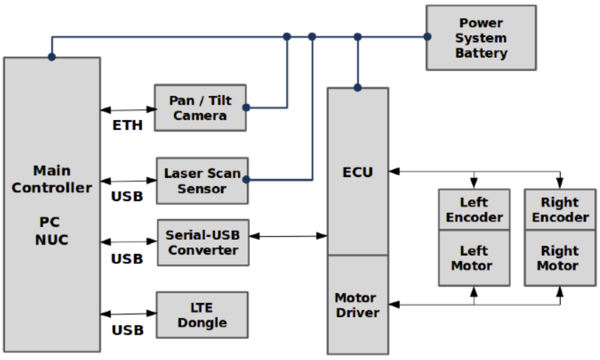

Fig. 2: Robot Hardware Architecture.

control and serial data transmission to the on-board CPU. The Serial-USB converter transforms serial data coming from the ECU board into USB data signals. Since real-time communication is a necessary prerequisite to ensure safe robot navigation, we rely on LTE technology, which is normally used for very large broadband communications, ensuring low latency communication. Our system accesses the Internet through a USB 4G LTE dongle. Core of the on-board processing is a mini PC NUC board DN2820FY with CPU Intel Celeron N2820 @2.1Ghz (dual-core). This mini PC exposes three USB ports that we use to gain access to laser scan sensor data, serial data coming from the ECU board and data from LTE dongle. A LAN Connector RJ45 is used to connect the mini PC with the Pan/Tilt Camera.

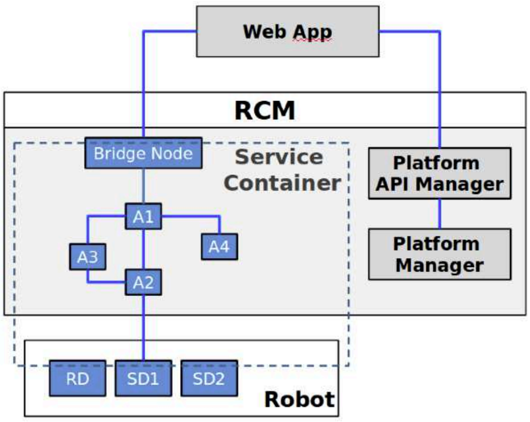

Fig. 3: Block diagram of the Cloud-based Robotics Platform.

B. Cloud Robotics platform Architecture

Proposed platform relies on the ROS framework in order to exploit the vast amount of already developed ROS nodes and service applications. Fig. 3 shows a schematic block diagram of the proposed CRP framework.

RCM generates a Service Container (SC) which hosts all ROS nodes needed to control the robot. These nodes perform the computation necessary to support the abstraction of hardware layer. We define 4 different types of ROS nodes:

- *Sensor Drivers (SD1, SD2)*: ROS nodes that abstract the information from the sensors (Laser scan and PT camera) and encapsulate them into ROS messages;
- *Robot Driver (RD)*: ROS node that abstracts status update from the robot and encapsulate them into ROS messages;
- *Application nodes (A1, A2, A3, A4)*: ROS nodes that process all the information coming from the sensors and the robot;
- *Bridge node*: ROS node that implements a Web-socket transport layer to connect the SC with the web application.

Basically, the web application knows nothing of the robot except the information that it publishes through ROS messages. As a meaning of fact, the platform creates a cloud abstraction of the physical robot, that we call the cloud-agent (or *robot clone*).

C. Web User Interface

End users can interact with the robotic platform using a simple but effective GUI (a screenshot is shown in Fig. 4), specifically design to maximize user experience. This GUI is composed by 5 main components:

- *Streaming video camera* (1): this component is located at the top left of the interface and it allows user to watch digital video stream from the camera;
- *Laser Image flow* (2): this component is located at the top right of the interface and it shows data stream from the laser scanner, useful to highlight and avoid obstacles;
- *Teleoperation buttons* (3): these elements are located at the bottom of the interface and are used to move the robot and the camera;
- *Assistance panel* (4): this panel is located at the bottom of the interface and provides information and assistance to the user during teleoperation. In particular, it provides meaningful information about the robot state, such as the autonomy left, speed, camera angle position and the distance from the nearest obstacle (both static or dynamic) detected.

III. Telepresence Service Application

In this Section, we focus on the description of the architecture of the proposed robot application. The considered use-case is the one of a telepresence robot that is teleoperated by a mobility impaired user to explore an otherwise inaccessible museum. The proposed solution allows the user to both follow predefined tour using autonomous navigation technologies and visiting in autonomy the whole museum.

Fig. 5 shows the functional architecture of the service application. The architecture is composed by three different

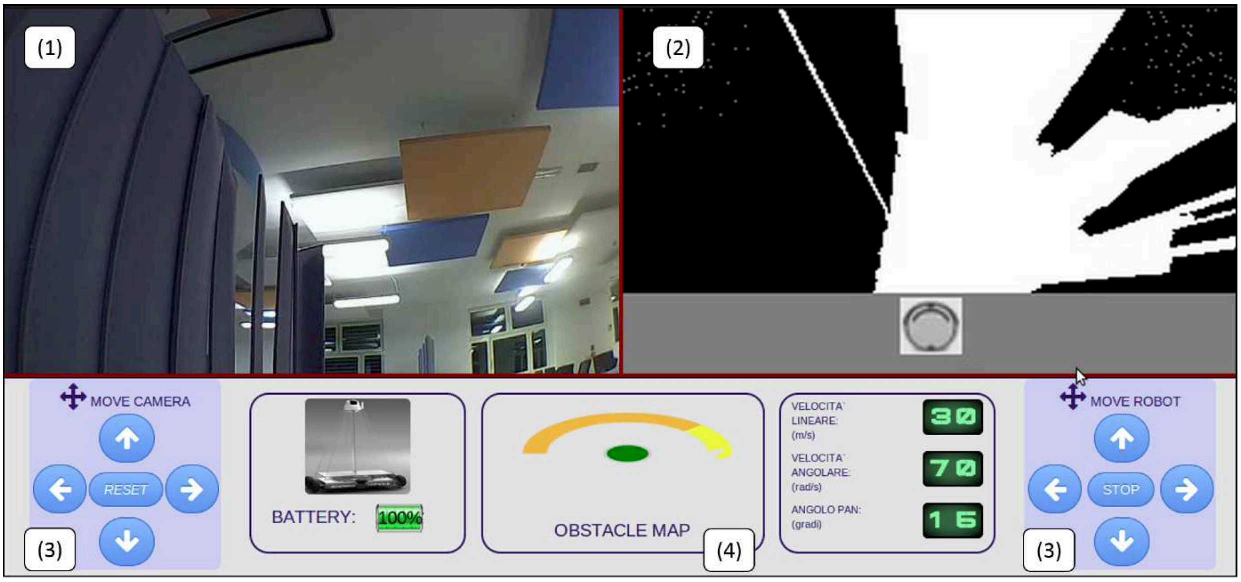

Fig. 4: Graphical User Interface (GUI).

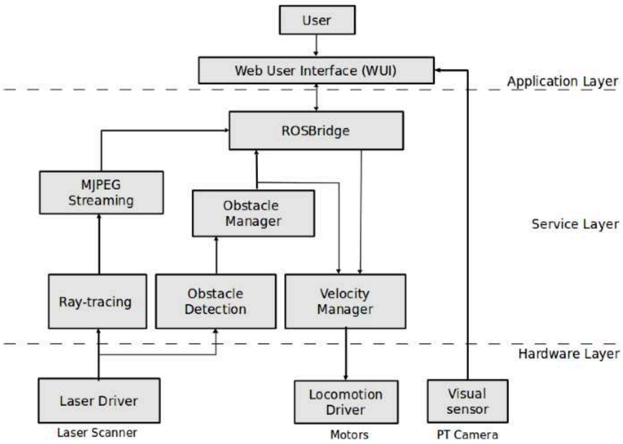

Fig. 5: Functional architecture of the assisted telepresence application.

layer: the *hardware layer*, the *service layer* and the *application layer*:

- the *hardware layer* represents the robot platform itself, including in particular robot locomotion (i.e., wheel motors), navigation sensors, such as laser scanner and wheel encoders, and visual sensor. Each element of the hardware layer is paired with the respectively driver that ensures ROS communication;
- the *service layer* groups all the ROS nodes that are devoted to control the service application. The *ROSBridge* block [20], in particular, is used to allow communication between ROS and the Web User Interface through a simple communication protocol based on JSON messages.
- the *application layer* represents the high level user interface, used by the end user to interact with the robot.

More specifically, the Laser Driver block is devoted to collect information from the laser scan sensor and to send them to the Service Layer. Ray-tracing block converts the Laser Scan data into a jpeg Cartesian image used to estimate the environment nearby the robot. Then the mjpeg Streaming Block compresses such image that is sent to ROSBridge to allow video streaming. Obstacle detection block reads information from the laser scan sensor and evaluates the nearest obstacles to the robot. It provide this information to the Obstacle Manager to generate warning signals when needed (e.g., the nearest obstacle is far less than 50 cm from the robot). Velocity Manager processes warning signals and commands sent by the user in order to let the robot move properly but safely. The Visual Sensor block sends video frames to the GUI through RTP streaming. Finally, ROSBridge node provides a Web-socket transport layer to connect the Service Layer with the Application Layer.

A. *Autonomous Navigation*

Autonomous navigation [21] is one the main features of the proposed system: in particular it aims at computing a feasible path in order to reach a position in the environment, while avoiding both static and dynamic obstacles. The Autonomous Navigation architecture is based on the ROS Navigation Stack that implements a two-stages path planning procedure, composed by a *global planner* and a *local planner*. The global planner computes a reference path in order to reach a given target point, assuming the map of static obstacles is known Once the global path is computed, it is followed until the local planner changes it to react to dynamic obstacles in the robot surrounding.

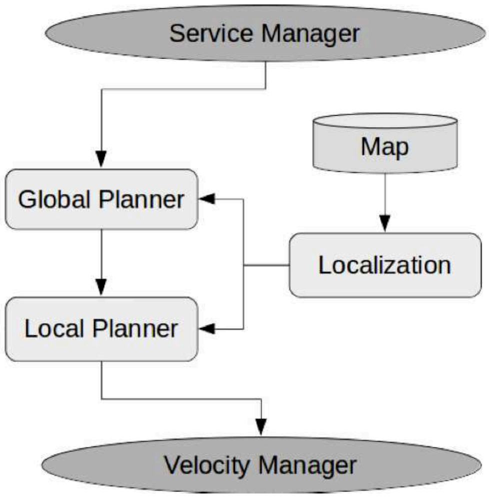

Fig. 6: Proposed architecture for autonomous navigation.

Both global and local planner require an estimation of the robot's position, that is computed by the Localization module. The map is stored in the Map Database, which contains all the maps that have been created for a given environment. At least a complete map of the environment is required for autonomous navigation; while navigation is running, map is updated by sensors when needed (e.g., if new static obstacles are detected).

Both planners communicates with the Service Manager and the Velocity Manager (as shown in Fig. 6). Service Manager is a module used to set the goal position and issue the stop command when target is reached, while the Velocity Manager module is used to define robot speed and direction, coherently with outcomes from the local planner.

The global planner relies on the *Optimal Rapidly-exploring Random Trees* (RRT*) [22], [23] algorithm. It is a reliable and fast sample-based algorithm that allows rapid exploration of any environment, providing a valid path to the goal target and avoiding any obstacles. The local planner is implemented using *Enhanced Vector Field Histograms* (VFH+) [24]. Localization is implemented using the Adaptive Monte-Carlo localization (AMCL), proposed in [25]. Monte-Carlo localization approaches recursively estimate the posterior probability of the robot's pose using particle filters (sample-based implementation of Bayesian filters). Chosen algorithm ensures a reliable and fast navigation, that could be even specifically optimized for a specific environment (e.g., in the case of archaeological sites where the robot should avoid more fragile areas).

IV. Preliminary Validation

We have tested the proposed system in a real museum environment in order to assess the feasibility of the whole idea and retrieve useful feedback of the platform.

Only the sensors drivers and Hardware Layer ROS nodes (see Fig. 5) were running onboard of the robot, while the remaining of the application logic was running on the cloud. In this early stage of validation our cloud is represented by a dedicated server equipped with an Intel Xeon processor and 4 Gb of RAM running ROS.

In particular, tests have focused on the capability of the robot to localize itself and navigate autonomously during normal daily activities, in presence of people visiting the museum.

Firstly, we evaluated the performances of the position tracking algorithm against a ground-truth map of the museum that has been drawn by hand. We measured the localization accuracy in several points inside the environment. Results shown that average errors are adequate for the subsequent path planning.

Secondly, we tested localization and path planning performances in the same environment with the previously cited map. A challenging path composed by more that one hundred waypoints was created. The robot was able to correctly localized itself and follow the given path, without missing any waypoint, even when moving in a particularly crowded area of the museum.

Finally, we evaluated platform response and network latency during remote teleoperation. In all the experiments, commands from the web GUI reached the robots in less than $10ms$, while the video stream from the robot camera was received and shown by the GUI in less than $5ms$, so that real-time teleoperation was achieved.

V. Conclusion

In this paper, we propose a novel robotic application based on a Cloud Robotics Platform intended to allow mobility impaired people to enjoy the whole museum experience. The Cloud Platform receives information from sensors mounted on top of the robot itself and computes appropriate commands to face with the given navigation tasks. We propose a Web User Interface that allows user to easily retrieve status update from the robot and send high level commands. In the proposed application, the robot can be directly teleoperated by the user. On the other hand, the Cloud Robotics Platform provide the robot of autonomous navigation capabilities that can be used to allow the user to give high level commands (e.g., reach a specific statue within the museum) instead of directly control the robot.

As shown by the early validation phase, the proposed solution is able to safety navigate within crowded museum environments. Future works will be devoted to improve interaction patterns with end users and to test the proposed application in others and more challenging real case scenario (e.g., partially agible museums and archaeological sites).

Acknowledgment

This research has been partially funded by the AsTech CINI Lab.

References

[1] G. Dudek and M. Jenkin, *Computational principles of mobile robotics*. Cambridge university press, 2010.

[2] D. Feil-Seifer and M. J. Mataric, "Defining socially assistive robotics," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*. IEEE, 2005, pp. 465–468.

[3] S. W. Brose, D. J. Weber, B. A. Salatin, G. G. Grindle, H. Wang, J. J. Vazquez, and R. A. Cooper, "The role of assistive robotics in the lives of persons with disability," *American Journal of Physical Medicine & Rehabilitation*, vol. 89, no. 6, pp. 509–521, 2010.

[4] G. Song and S. Guo, "Development of a novel tele-rehabilitation system," in *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*. IEEE, 2006, pp. 785–789.

[5] T. Mouri, H. Kawasaki, T. Aoki, Y. Nishimoto, S. Ito, and S. Ueki, "Telerehabilitation for fingers and wrist using a hand rehabilitation support system and robot hand," in *Proceedings of the 9th International IFAC Symposium on Robot Control (SYROCO'09)*, 2009, pp. 751–756.

[6] M. Cortese, M. Cempini, P. R. de Almeida Ribeiro, S. R. Soekadar, M. C. Carrozza, and N. Vitiello, "A mechatronic system for robot-mediated hand telerehabilitation," *Mechatronics, IEEE/ASME Transactions on*, 2014.

[7] A. Bonarini, S. Ceriani, G. Fontana, and M. Matteucci, "On the development of a multi-modal autonomous wheelchair," *Medical Information Science Reference (an imprint of IGI Global), Hershey PA*, 2013.

[8] G. A. Farulla, L. O. Russo, C. Pintor, D. Pianu, G. Micotti, A. R. Salgarella, D. Camboni, M. Controzzi, C. Cipriani, C. M. Oddo *et al.*, "Real-time single camera hand gesture recognition system for remote deaf-blind communication," *Augmented and Virtual Reality*, pp. 35–52, 2014.

[9] F. Michaud, P. Boissy, D. Labonte, H. Corriveau, A. Grant, M. Lauria, R. Cloutier, M.-A. Roux, D. Iannuzzi, and M.-P. Royer, "Telepresence robot for home care assistance." in *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*. California, USA, 2007, pp. 50–55.

[10] 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," *Telemedicine and e-Health*, vol. 13, no. 4, pp. 407–424, 2007.

[11] L. Tonin, T. Carlson, R. Leeb, and J. del Millan, "Brain-controlled telepresence robot by motor-disabled people," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 4227–4230.

[12] K. Walker, "Designing for meaning making in museums: Visitor-constructed trails using mobile digital technologies," Ph.D. dissertation, Institute of Education, University of London, 2010.

[13] L. Tallon. (2009) Learning times museum handheld survey 2009. [Online]. Available: http://www.learningtimes.com/news/museum-handheld-survey-2009/

[14] G. Mohanarajah, D. Hunziker, R. D'Andrea, and M. Waibel, "Rapyuta: A cloud robotics platform," 2014.

[15] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," 2015.

[16] 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, no. 3.2, 2009, p. 5.

[17] G. Ermacora, S. Rosa, and B. Bona, "Sliding autonomy in cloud robotics services for smart city applications," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*. ACM, 2015, pp. 155–156.

[18] G. Ermacora, A. Toma, S. Rosa, and R. Antonini, "Leveraging open data for supporting a cloud robotics service in a smart city environment."

[19] S. Rosa, L. O. Russo, and B. Bona, "Towards a ros-based autonomous cloud robotics platform for data center monitoring," in *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*. IEEE, 2014, pp. 1–8.

[20] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: Ros for non-ros users," in *Proceedings of the 15th International Symposium on Robotics Research*, 2011.

[21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[22] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt*," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1478–1483.

[23] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[24] I. Ulrich and J. Borenstein, "Vfh+: Reliable obstacle avoidance for fast mobile robots," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2. IEEE, 1998, pp. 1572–1577.

[25] D. Fox, "Kld-sampling: Adaptive particle filters," in *Advances in neural information processing systems*, 2001, pp. 713–720.