UChile Peppers 2019 Team Description Paper

Cristopher Gómez, José Astorga, Nicolás Marticorena, Lukas Pavez, Rodrigo Salas, and Javier Ruiz-del-Solar

[1] Department of Electrical Engineering – Advanced Mining Technology Center (AMTC)
Universidad de Chile
`http://robotica-uchile.amtc.cl/pepper-index.html`

Abstract. The UChile Peppers robotics team participates in the RoboCup @Home Social Standard Platform League (SSPL) since its creation in 2017. In the 2018 RoboCup World Competition the team obtained the third place. The main contributions of the team are the use of a ROS system completely embedded in Pepper's computer, the implementation of a state-of-the-art Visual-SLAM system on Pepper, and the development of a backpack to carry additional computational hardware for the robot. Moreover, it is worth mentioning that the team makes public its developments on Github.

Keywords: Visual SLAM, Social Robots, Social Standard Platform League.

1 Introduction

The Pepper robot presents a novel challenge to the @home community. This robot is very attractive to humans thanks to its friendly appearance and lifelike movements. However, its range sensors and computer have several limitations when compared to other robots of the @home OPL. This situation poses several challenges for the correct resolution of @home tests.

The UChile Peppers team is part of the @Home SSPL since its creation in 2017. The developments of the team are focused on developing systems that can run entirely on Pepper's onboard computer. To collaborate with other teams and to share code and designs are also part of the team's goals. All developments made by the team are available through GitHub.

2 Background

The UChile Robotics Team (UChileRT) has been involved in RoboCup competitions since 2003 in different leagues: Four-legged 2003-2007, Standard Platform League (SPL) in 2008-2018, Humanoid League in 2007-2009, @Home in 2007-2015, @Home SSPL in 2017-2018, and @Home OPL in 2017. Moreover, Prof. Javier Ruiz-del-Solar was the organizing chair of the Four-Legged competition in 2007, TC member of the Four-Legged league in 2007, TC member of the @Home league in 2009, Exec Member of the @Home league between 2009 and 2015, and co-chair for the

RoboCup 2010 Symposium. Among the main scientific achievements of the group to be highlighted, are five important RoboCup awards: RoboCup 2004, 2015 and 2017 Best Paper Award, and RoboCup @Home Innovation Award in 2007 and in 2008. In addition, UChileRT's team members have published a total of 46 papers in RoboCup International Symposia between 2003 and 2018, 21 of them corresponding to oral presentations.

In the SPL league where Nao robots are used, UChileRT reached fourth place in three consecutive RoboCup World Competitions (2014 in Brazil, 2015 in China and 2016 in Germany).

In previous versions of @Home SSPL competitions, the UChile Peppers team reached the fifth place among seven teams in 2017, and the third place among nine teams in 2018.

3 Current research

3.1 Visual SLAM

Pepper’s LIDARs have a low resolution and short range [1] compared to the common LIDARs used on @Home OPL [2]. This fact hinders self-localizing in large spaces with walls that are farther away from the LIDAR range (Fig. 1). Furthermore, the low resolution of the lasers makes difficult to differentiate between a wall and movable furniture. For these reasons, a LIDAR-based localization system does not provide reliable results for Pepper operation.

As an alternative for the classic LIDAR-based localization, we proposed a Visual SLAM approach this year [3]. ORB-SLAM [4][5], a state-of-the-art system characterized by its efficiency, re-localization capabilities and a lightweight localization-only mode is modified to solve its main issues: unknown scale factor and scale drift. ORB-SLAM provides a system to be used with stereo, RGB-D and monocular cameras. At first, we work with monocular ORB-SLAM that suffers from the unknown scale factor problem. To solve this, odometry information is added to the estimation procedures (Fig.2).

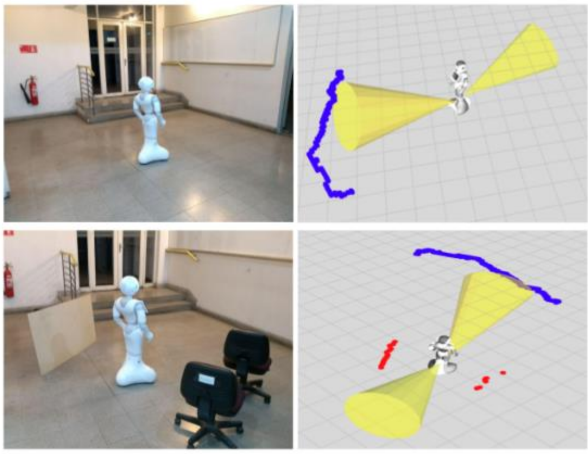

Fig. 1. Pepper's sensors visualization. Blue corresponds to a simulated Laser Scan from depth images. The sonars are represented by the yellow cones. The LIDAR (red) does not register measurements in the first scenario because the walls are out of range. In the second scenario, the readings for a wall and a pair of chairs are very similar.

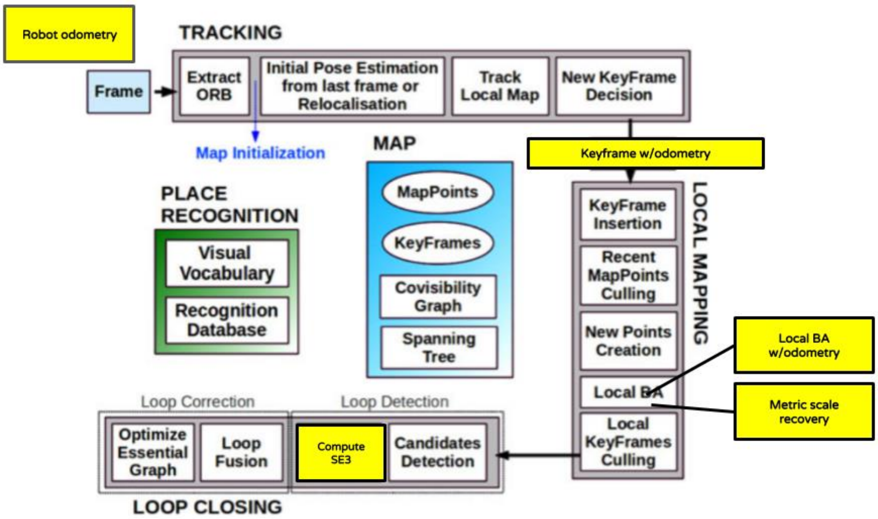

Fig. 2. ORB-SLAM modified software diagram. The yellow blocks correspond to the modifications. First, odometry information is added as an input together with the frames. Second, all the Keyframes have an odometry measurement associated.

ORB-SLAM receives a frame from a camera. This frame is processed to extract ORB features. By matching features from different sequential frames, it is possible to build a map and calculate the camera position in this map. However, as mentioned early this map lacks a metric scale. To give the map a metric scale, the robot odometry measurements are given together with the frame. The odometry is then used to calculate a scale factor that is applied to the map. Additionally, all map optimization procedures consider the odometry measurements as a variable, giving robustness to drift phenomena.

In addition to ORB-SLAM base software modifications, a group of improvements were made to use ORB-SLAM alongside ROS Navigation Stack [6]. First, a save and load map functionality was added. Second, an additional ROS Node was developed to give a transformation between a map and Pepper, based on information provided by ORB-SLAM and Pepper's kinematic information. This enables a direct bridge between our Visual SLAM localization solution and the **move_base** ROS package (Fig.3).

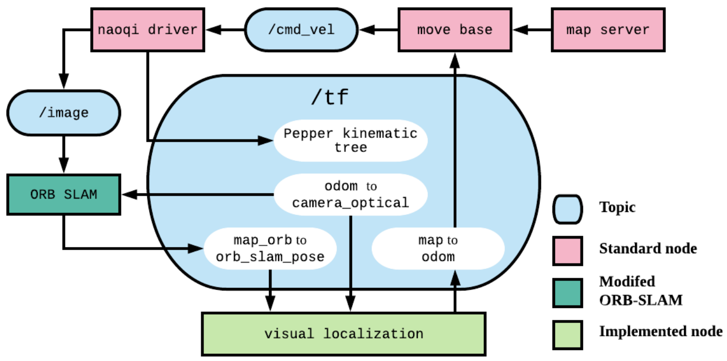

Fig. 3. Software diagram for the localization and navigation system. ORB-SLAM is connected to the standard ROS Navigation System.

3.2 Pepper Backpack

To enhance Pepper perception abilities without sacrificing speed nor mobility, a custom backpack was developed [7]. The backpack can carry a Nvidia Jetson TK1 single-board computer together with a battery. The backpack can be easily attached to Pepper through non-invasive means (Fig. 4). Through a high-speed Ethernet Gigabit connection, Pepper can stream images to the Jetson TK1, so they can be processed with state-of-the-art deep-learning based algorithms. As a proof of concept, object detection using tiny-YOLO [8] was successfully implemented with an inference rate of 10 Hz.

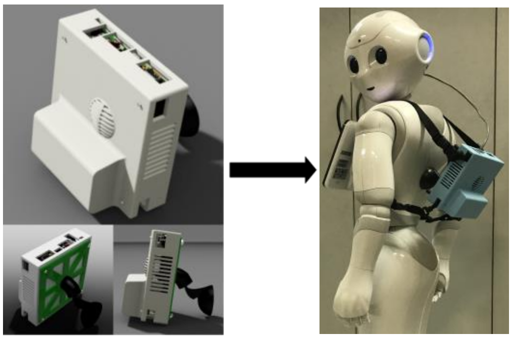

Fig. 4. The Pepper's Backpack is a simple 3D printed enclosure for Nvidia Jetson TK1 with space for a battery. It can be easily attached to Pepper without impairing his movements.

3.3 Long-Term Memory for Social Robots

Memory relates to the storage of experiences in the brain [9]. It can be divided in Short-Term Memory (STM) and Long-Term Memory (LTM). Although Short-Term Memory manages very detailed information, it has little storage capacity. Long-Term Memory takes cares of a large amount of information about experiences and entities. However, it has less detail and takes more time to access the information. An explicit and implicit memory composes LTM [9]. The first is further classified in episodic memory and semantic memory. Episodic memory refers to memories of specific events. Semantic memory corresponds to the knowledge of the world. Long-Term Memory is necessary to develop robots that can really have a natural interaction with humans. Memory gives continuity and identity.

A first implementation of LTM for service robots was presented in [10]. The system can store, acquire, and manage episodic information. Additionally, it has a simple process of memory consolidation. However, is limited only to two semantics entities: persons and objects. Also, it lacks a standard API that enables a smooth integration within a robotic system. A new system is being developed by our team to solve these issues [11]. This new LTM system has the following characteristics:

- Semantic and episodic memory focused on service robot's tasks
- Server for continuous acquisition of information
- Implementation based on ROS framework

4 Software Architecture

The main framework used to develop software for Pepper is ROS (Robot Operating System). The native framework NAOqi provided by Pepper's developer, Softbank Robotics is used indirectly. A bridge between ROS and NAOqi enables the access to Pepper's sensors and actuators through a standard ROS interface. In addition, NAOqi provides some useful systems like Speech Recognition, Sound Localization, Face Detection and Recognition, Age and Gender Recognition, Person Detection, etc.

One of the principal reasons for using ROS as the main framework is that we can use all the software already developed for ROS compatible machines, especially those that were developed for our Bender robot of the @home OPL. The diagram of Fig. 5 shows the relationships between the ROS software and the Pepper robot. Another plus coming from using ROS is the option to use the ROS-Python Skill interface developed in our laboratory. This interface provides an easy way to program the high-level behaviors. This Python Interface is heavily inspired by Tech United @Home team own interface [12]. Additionally, the ROS-Python Skill interface grants the option to share behaviors between our Bender robot of OPL and our Pepper robot of SSPL.

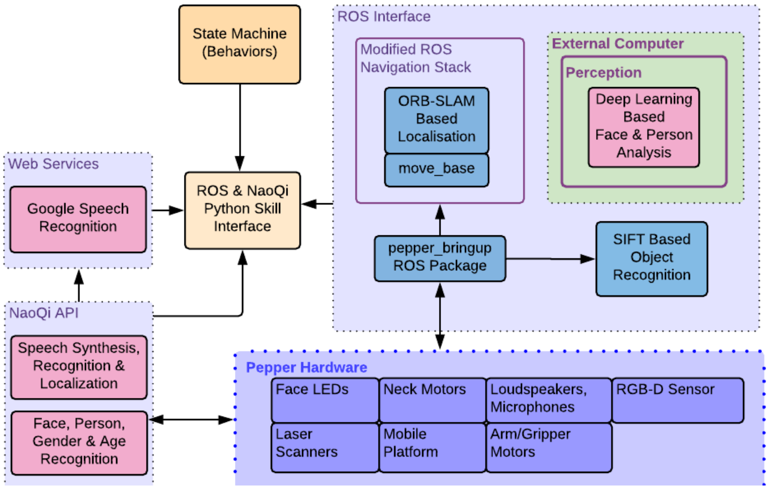

Fig. 5. Pepper main software diagram

The ROS framework has been successfully installed on the Pepper's onboard computer. The challenge to install ROS in Pepper could be very time consuming for new teams. The Operating System running on the onboard computer is an old version of Gentoo without a good package manager and without native support from ROS. Therefore, the easiest option is to compile ROS and its dependencies directly from the source code. A list of library dependencies can be found in our Github page.

All the necessary system to run Pepper as a safe service robot runs inside its onboard computer. However, because of the limited computational power of Pepper, the perception systems are less accurate than the ones used in the @home OPL. This can be solved with external computation. Nevertheless, in our case external computation is only used for non-critical systems. For example, navigation, as a critical system, runs entirely on Pepper's internal computer. This means that even if the wireless connection is lost, the robot can still avoid collision with objects or people.

5 Re-usability of the System for other Research Groups

All our developments are publicly available through our Github. This includes our developments on Visual SLAM (who are mainly ORB-SLAM modifications), and the instructions to build a Pepper's Backpack.

Based on the interest in the Pepper's Backpack shown by other @home SSPL teams and people from SoftBank, the team is working toward new versions for the Backpack aimed at different onboard computers.

6 Applications in the real world

The UChile Peppers team is composed of undergraduate students. The team organizes workshops to teach robotics to students at our university.

Moreover, the Pepper robot is one of an important tool used by the university in its efforts to promote and disseminate science and technology to general population, especially the young. The team participates in talks and STEM exhibitions along the year (Fig. 6).

Fig. 6. Left, Pepper in an open laboratory exhibition. Right, Pepper in a presidential act.

7 Conclusions and future work

This Team Description Paper presents the advances made by the UChile Peppers team during the last year. Additionally, it refers to some plans for future developments based on RoboCup@home goals.

The main developments are focused on narrowing the gap between Pepper and other commercial and custom robots, which have more processing power and better sensors. Currently, the most pressing subject is localization and navigation. To this end, a Visual SLAM approach has been developed.

Because one of the strong areas of our laboratory are Deep Neural Networks (DNNs) for computer vision, the team is investigating novel implementations of compression and quantization of DNNs to be used in Pepper.

Finally, once the basic abilities of Pepper are reliable enough, the team wants to develop real-life applications like using Pepper as a sight guide for people with impaired vision.

8 Acknowledgments

This work was funded by FONDECYT Project 1161500. Our work is supported by NIC Chile.

References

1. Perera, V., Pereira, T., Connell, J., Veloso, M.: Setting Up Pepper For Autonomous Navigation And Personalized Interaction With Users. (2017).
2. Wachsmuth, S., Holz, D., Rudinac, M., Ruiz-Del-Solar, J.: RoboCup@Home -- Benchmarking Domestic Service Robots.
3. Gómez, C., Mattamala, M., Resink, T., Ruiz-del-Solar, J.: Visual SLAM-based Localization and Navigation for Service Robots: The Pepper Case. In: RoboCup 2018: Robot World Cup XXII. , Montreal (2018).
4. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans. Robot. (2015).
5. Mur-Artal, R., Tardos, J.D.: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Trans. Robot. 33, 1255–1262 (2017).
6. Eitan Marder-Eppstein: ROS Navigation Stack, http://wiki.ros.org/navigation.
7. Reyes, E., Gómez, C., Norambuena, E., Ruiz-del-Solar, J.: Near Real-Time Object Recognition for Pepper based on Deep Neural Networks Running on a Backpack. In: RoboCup 2018: Robot World Cup XXII. , Montreal (2018).
8. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. 2017–Janua, 6517–6525 (2017).
9. Eichenbaum, H.: Memory. Scholarpedia. 3, 1747 (2008).
10. Sánchez, M.-L., Correa, M., Martínez, L., Ruiz-del-Solar, J.: An Episodic Long-Term Memory for Robots: The Bender Case. In: Almeida, L., Ji, J., Steinbauer, G., and Luke, S. (eds.) RoboCup 2015: Robot World Cup XIX. pp. 264–275. Springer International Publishing, Cham (2015).
11. Pavez, M., Ruiz-del-Solar, J., Amo, V., Driehausen Meyer zu, F.: Towards Long-Term Memory for Social Robots: Proposing a New Challenge for the RoboCup@Home League. In: RoboCup 2018: Robot World Cup XXII. , Montreal (2018).
12. Burgh, M.F.B. Van Der, Lunenburg, J.J.M., Appeldoorn, R.P.W.: Tech United Eindhoven @ Home 2018 Team Description Paper. (2018).
13. King, D.E.: Dlibml: A Machine Learning Toolkit. J. Mach. Learn. Res. 10, 17551758 (2009).
14. Amos, B., Bartosz, L., Satyanarayanan, M.: OpenFace: A general-purpose face recognition library with mobile applications. School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213 (2016).
15. Hassncer, T., Levi, G.: Age and gender classification using convolutional neural networks. 2015 IEEE Conf. Comput. Vis. Pattern Recognit. Work. 34–42 (2015).
16. Luca, M., Bellavista, D.: Recognizing Emotional States in Faces., https://github.com/luca-m/emotime.

Robot's Description

Pepper's hardware description is standard and can be found on the official documentation provided by SoftBank.

Regarding software, the SoftBank supported ROS packages provide the core features to work with the robot (drivers and URDF model of Pepper). Furthermore, we have developed an improved version of the simulated robot for Gazebo, on top of the baseline provided by SoftBank.

Internal Computer:

- Navigation: ROS Navigation stack is used for mapping, localization and planning. A Visual SLAM based localization system is also used.
- Face detection/recognition: NAOqi systems for Face detection and recognition are used.
- Facial features recognition: NAOqi provided system for age and gender recognition.
- Speech recognition and generation: We use the already incorporated speech recognition and generation system in the robot.
- Sound Localization: NAOqi provided sound localization system
- Object Recognition*: SIFT/based object recognition system.

External Computation:

- Face detection/recognition: For detection we use the **dlib** library [13]. For recognition, we use the Openface research [14] based on the framework Torch.
- Facial features recognition: The following facial features have been developed: age and gender recognition [15] using CNN with the framework Caffe, and emotion recognition with the **emotime** library[16].
- Arms control and two-hand coordination: The ROS MoveIt! Package is used and has been successfully tested in the simulated version of Pepper.
- Google Speech Recognition

The implementation of high-level behaviors is achieved with hierarchical state machines, programmed with the **smach** library of Python. An emphasis is given to modularity and reutilization of code. Additionally, to simplify the programming of high-level behaviors, a middle layer between the state machines and the ROS control interface exist based on the **robot_skill** interface designed by Tech United Eindhoven @HOME team.