Getting State with RoboFEI Judith for RoboCup@Home 2017

Andrey Aurora Masiero, Igor dos Santos Soares, Marina Yukari Gonbata, Leonardo Contador Neves, Thiago Spilborghs Bueno Meyer, Rafael Yuji Ueda Yamamamoto, Flavio Tonidandel and Plínio Thomaz Aquino Junior

Centro Universitário FEI, São Paulo, Brazil,
http://www.fei.edu.br/robofei

Abstract. This team description paper (TDP) presents a robotic project from RoboFEI team to RoboCup@Home competition on RoboCup Competition 2017. We use PeopleBot platform as base to develop our final robot. We attached Microsoft Kinect, camera and LIDAR (Light Detection And Ranging) sensor to improve robot capabilities. The chosen robot was the PeopleBot, a manipulator-free robot, combined with the Microsoft Kinect. In the development core the system used was the Robot Operating System (ROS), accompanied by the Application Programming Interface Open Natural Interaction (API OpenNI) for movement and gesture recognition, and for the speech and voice recognition the Google Speech Web was manipulated. In addition, a manipulator was inserted to the PeopleBot with the intention of accomplishing works such as moving and carrying objects.

1 INTRODUCTION

Today, the concern that surround the helping services, such as helping human being in domestic and personal ambient, and assistive way of using technology has increased. With the purpose of serving this necessities, in 2006 the Robot Competition known as RoboCup@Home was created. This is a Multi-task competition, which has as main areas involved, the Interaction and Cooperation Human-Robot, Computer Vision, Patterns Recognition, Behavior Analysis, Artificial Intelligence and other areas [1].

Thus, to participate in the RoboCup@Home competition it is required an autonomous mobile robot development. Usually, the participant robots have manipulators to execute works that a human would do using its hands and arms. Wheels are usually responsible for the movement, since it is more stable than legs, used in the humanoids robots [1].

With this scenario in mind, the ROBOFEI @Home crew is going to use the PeopleBot robotic platform to participate in the competition. However, the PeopleBot doesn't have , in it's original *setup*,manipulators, making it impossible to accomplish the required tasks in the competition. The tasks that don't involve manipulators, on the other hand, can be achieved with the robot in its actual

setup. In this Team Description Paper (TDP) the environments and the equipments used in the robot configuration will be presented, along with some tasks that it can realize [1],[2].

The rest of the TDP is organized in this form: in section 2 the control and robot development, ROS, is presented. In the 3.1 section some of the sensors are described and *frameworks* utilized in this work. The 3.1 section shows the API, which makes possible for the robot to recognize voices. In the 8 section, the robot project and some possible tasks that it can achieve are described, and the last section, 9, cease the TDP.

2 DEVELOPMENT ENVIRONMENT ROS

NROS (*Robotic Operating System*) is a *framework* which objective is to simplify the *software* development for robots. it has tools and libraries to realize complex and simple tasks, available for many robotic platforms. ROS first version, released under the *framework* structure, occurred in January 2009, with the Mango Tango version.Since then, various distributions were released, and the last one, named Indigo Igloo, released in July 2014 [3].

Beyond the robot *hardware* manipulation librabries, such as PeopleBot, and also robot behavior controlling libraries, ROS is equipped with simulating tools. This simulators works with all robots in every pre-mapped and even on environments mapped by robots in real aplications. This tools transforms ROS, nowadays, in one of the most used *frameworks* in robot programming. It's possible to count with many libraries in the market, such as OpenCV, OpenNI, NITE and others. Due to this facts, it was opt to use ROS in the software development for RoboCup@Home competition, being now the heart of PeopleBot application [3].

3 ROBOT VISION

Being able to recognize who are you talking to or which object are you looking at is an important part of our daily needs and when we talk about robotics, it is as important as it is for the humans. Many researchers are developing several ways to process and do a real-time recognition of what the robot sees through the camera. In the ROBOFEI @home project we utilize two image sensors, the Kinect sensor and the camera sensor each of them for different tasks, we will talk more about them in the next sections.

3.1 KINECT SENSOR

The Kinect is an equipment developed by Microsoft, which gained popularity when it was released with the XBOX 360 video game [4]. It, actually, is characterized by a group of sensors helping the performance presented by the Microsoft product. Its components are a RGB camera, a infrared sensor, which

makes possible to calculate depth, and built-in microphones allowing to detect several persons in one environment. The whole Kinect set have a processor and a proper software, independent of the video game or computer, moreover it is able to capture 48 articulation points in human body [4].

All specifications presented until now makes the Kinect an excellent sensor, not only for video games, but also for research, since its cost-benefit relation is very high. Since its release, in November 2010, many enthusiasts and researchers communities began to create frameworks to use the Kinect with other products [4]. Some of this frameworks are already inserted in ROS [3], they are:

- Kinect SDK
- OpenNI
- NITE
- Freenect
- OpenKinect

Each framework has a peculiarity such as, performance plataform, objectives and even competition. In the ROBOFEI @Home project the OpenNI [5],[6] was used, with the objective of accessing the main functions of the Kinect in several platforms, and the NITE, responsible for people tracking functions. Both will be presented with more details in the next section.

OPENNI AND NITE OpenNI is an API set that assists the application development which needs voice and hands gestures recognition and human body movements trackng. The API set is focused on, specially, on sensors like Kinect. In april 2014, Apple ended the OpenNI project, after the Primesense enterprise acquisition which was the supporter of the framework. Despite the project ending, some enterprises and Third-repositories saved the binary archives, so it could be possible to use it in new works [5],[6].

Inside OpenNI there is an API dedicated to human body tracking. It is named NITE. Through this functions is possible to attend basic tasks in RoboCup@Home, such as, for example, follow a certain person. [5],[6],[1].

3.2 CAMERA SENSOR

Due to the low Kinect's camera resolution, we cannot use it to the recognition tasks, so we use the Logitech C920 web cam. It contains a full HD camera and built-in dual stereos microphones. As our goal is to recognize objects and faces, we need a high resolution camera, so we can extract as many features as we can to achieve our goal and do a proper recognition of what the robot is looking at. For the recognition task in the ROBOFEI @home project was used the OpenCV, with the goal of finding objects and people in the image, our next section will present more details about this library.

OpenCV OpenCv is a image processing library, originally developed by Intel, and now maintained by Itseez, it was built to support people providing optimized computer vision algorithms. There are several algorithms in this library, from face and object detection to camera movements tracker algorithms. It is (highly recommended and utilized) in companies and research groups as the OpenCV is a BSD-licensed product, making it easy to use and modify its structure as needed.

4 VOICE RECOGNITION

Interactions through voice command are very important for human and robots to realize daily tasks and interact with each other. The team decided to use the cmu PocketSphinx speech recognition tool, which uses dictionaries and language models as base. The PocketSphinx tool is an offline tool based on the Sphinx model and the gstreamer. Its recognizer was used along with our codes, resulting in an acceptable trustworthy, within 60% and 99%. [7]. With one of the most important parts of interaction now described, in the next sections the whole RoboFEI @Home configuration for the competition will be presented.

5 ROBOT NAVIGATION

An autonomous robot, to be able to navigate alone, without the person help, it has competence to map the place where it is, to define its position in the space and it can decide which is the best possible route. To make this possible, sensors that capture external environments are used, after that this information is transformed in interpretable data for the robot and with these parameters it can choose the best route.

When the robot is in a unknown, it must do the environment, where it is located, mapping and at the same time define its position in the space. This technique is known as Simultaneous Localization and Mapping (SLAM). In the navigation, the robot has the capacity to choose the best possible route and avoid possible obstacles. For this to happen, it determinate parameters where there is the slightest path error and the robot is constantly correcting it.

The PeopleBot will use the laser sensor Hokuyo URG-04LX-UG01 to create the environment 2D map where it is located, and find possible obstacles in the course. This device is used to scan areas, it has a 4 meters range and a 240º angular degree. Emitting an infrared light with a 0.36º pace, the number of scanned point is equal to 683 points.Through this it can locate new obstacles that weren't on the map. An alternative to Hokuyo URG-04LX-UG01 sensor (2-dimensions data) is the point cloud data from Microsoft Kinect sensor. The point cloud data can describe the environment by a 3-dimensions information, thus the mobile robot may be able to detect a greater number of obstacles and the best way paths.Our robot uses a line data information (2-dimensions data) from Kinect point cloud too, providing two plans informations (Kinect and Hokuyo data) in a different heights of the mobile robot for getting informations about

obstacle that avoids from Hokuyo data like a tables, chairs or objects at a greater height.

6 MANIPULATORS

On the first part of development, some sketches were made with the main objective of creating a manipulator that would have the same number of Degrees of Freedom (DOF) contained in a human arm, with the purpose it obtaining a great similarity to real movements using the anthropomorphic principle.

Basing on that, an anatomy and human kinesiology study was initiate, more specifically on the skeleton of the free portion of the upper limbs, which are: arm, forearm, carpus, metacarpus. It was noticed that the main movements are extension and flexion.

Based on this, in the project second phase, the prototype was built using Computer-Aided Design (CAD) tools, which allowed the kinematics of movements and functional requirements research, and if any member failed during the simulation, the project would be redesigned. This way, creating an iterative creation process, with the intention of improving the prototype, as show on the picture. 1.

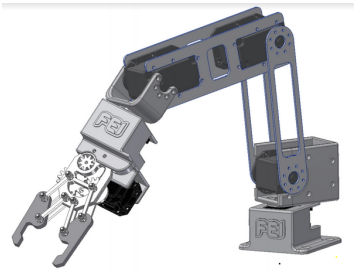

Fig. 1. Manipulator developed for the PeopleBot robot.

With the geometry analysis of each component, the most appropriate manufacturing process: for the parts with more complex shapes 3d printers were used and; for plain shape parts we preferred to use an aluminum production process, with results in a great resistance with less weight and a smaller dimension.

6.1 THE MECHANIC

For the manipulator movement, one Dynamixel RX-24F actuators, two RX-28 actuators and three RX-64 actuator are used, which are distributed among shoulders (one RX-64 and one RX-28F), elbow (one RX-64) and handle (one RX-64, one RX-28 and one RX-24F). Both actuators have 300º spin amplitude and are powered by 12V the RX-24F actuators, and with 18V the RX-28F and

the RX-64 actuators. Using a movement system for more than one paddle flow, ate the same time, reducing the number of servo motors.

6.2 THE KINEMATIC AND CONTROL

For the manipulator mobility some distinctive algorithms, which describe the freedom that it has in the space where it is located, are used. Manipulator extensors and junctions analysis methods that use direct and inverse kinematic were implemented, this way allowing to describe the behavior and state in the space. Together with an control algorithm, we can accomplish simple and even complex tasks, as the space informations can be captured through the sensors and comprehended, so it can make decisions for new tasks and interactions using the manipulator. Our Dynamixel actuator provides a internal micro-controller that sends some informations about the state of motor. These informations, like the speed and goal error are used into control algorithm providing more natural movements once more complex movements can be executed. A easily way example to get more natural movements with kinematic control is through attributing different speeds for each DoF of arm when it was in point A and going to point B, these different speeds are proportional with distances that each joint will course.

7 SMACH

SMACH is a ROS-independent Python library to build hierarchical state machines. The autonomous concept in robotics it's connected to create a knowledgeable robot able to solve any unknown problems in real word by itself without pre-programmed algorithms. SMACH may help to building a series of states or abilities that the robot has and link this abilities to execute a complex tasks autonomous. Like a human with states to walk or to stop walking in the robot with SMACH architecture is the same concept, the skills (or the states the robot has) are grouped and linked to build state machines which will create the knowledge of the robot, like a human states to walk,to talk or get something.

The greater advantage to use SMACH is the segregation of the algorithms that now are separated by skills or states and each state is independent of each other. SMACH gives through the smach_viewer a full introspection in your state machines, state transitions, data flow, etc. Smach_viewer has a graphic interface that allows the user to see what is going on with the robot in real time, including acts, problems or changes in environment variables.

8 ROBOFEI@HOME PROJECT

In the RoboFEI@Home the PeopleBot robotic plataform was used. The PeopleBot moves in the enviroment using its wheels located in the base, and it has sonars for the collision control in low places, for example a coffee table in the

living room. In its original configuration. there is also a RGB video camera in center of its top, which is also occupied by sonars. In the configuration for the RoboCup@Home competition, the camera available in the original configuration will only be used as an assistant, if it's needed.

For the robot vision in the competition, a Kinect sensor was added to the top of the robot. This way, it's possible to use not only the RGB camera, but also the depth sensors and microphones utilized for the communication with the Google Speech API. So, the sensors located in the robot top are not used in any task neither in PeopleBot control.

The task developed primarily for the this TDP construction was to follow a specific person without the interference of the people in the surroundings. The process is realized by going through the following steps:

- A person stands still in front of the PeopleBot;
- The robot performs a face recognition and register a digital skeleton in the memory;
- The Person asks for the robot to follow her/him;
- The robot starts the process and only stops when when commanded.

9 CONCLUSION

The robotic platform PeopleBot had a good performance in accomplishing the RoboCup@Home competition tasks. There are still some improvements to be made, so the mobile robot will be able to accomplish all the tasks purposed in the international competition. The first challenges were concluded with success and the robot autonomy is acceptable for the time requested to accomplish the tasks. In the future, the speech recognition will be improved, being able to even map the sound sources in the environment, and with this, researches and implementation of the remaining tasks expected in the RoboCup@Home manual will occur.

References

1. Robocup@home. http://www.robocupathome.org/. last access: May, 10th 2017.
2. Kai Chen, Dirk Holz, Caleb Rascon, Javier Ruiz des Solar, Amirhosein Shantia, Komei Sugiura, Jörg Stückler, and Sven Wachsmuth. Robocup@home 2017: Rule and regulations. http://www.robocupathome.org/rules/2017_rulebook.pdf, 2017.
3. Ros.org — powering the world's robots. http://www.ros.org/. last access: May, 10h 2017.
4. Kinect for windows. http://www.microsoft.com/en-us/kinectforwindows/. last access: August, 14th 2014.
5. Openni - wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/OpenNI. last access: August, 14th 2014.
6. Openni 2 downloads and documentation — the structure sensor. http://structure.io/openni. last access: August, 14th 2014.
7. Cmu pocketsphinx speech recognition. https://www.cs.cmu.edu/~awb/papers/ICASSP2006/0100185.pdf. last access: March, 10th 2017.

Robot Judith Hardware Description

(In this section briefly describe the software and hardware of the robot)

Robot Judith from RoboFEI specifications are as follows:

- Peoplebot Platform.
- Manipulator with 6 DOF and parts 3D printed.
- Microsoft Kinect for Xbox 360.
- Front and Rear SONAR Arrays and Front-Facing Upper SONAR Array.
- Hokuyo URG-04LX-UG01 laser.
- PS Eye microphone array.
- Yoga HT-320A microphone.
- Logitech C180 Web-cam.
- Galaxy Tab 7”

Fig. 2. Robot Judith

Robot’s Software Description

For our robot we are using the following software:

- Platform: ROS Indigo on Ubuntu 14.04 OS.
- Navigation, localization and mapping: SLAM
- Face recognition: LBP Algorithm
- Speech recognition: CMU PocketSphinx
- Object recognition: SIFT and SURF OpenCV Algorithms.
- Arms control: Inverse Kinematic implemented on ROS own packages.

All code are available on: `https://github.com/OpenFEI/rfh_judite`