

RoboFEI@Home 2016 Team Description Paper

Andrey A. Masiero, Douglas De Rizzo Meneghetti, Leonardo Contador,
Lucas Vasconcelos, Flavio Tonidandel and Plinio T. Aquino Junior

Centro Universitário FEI,
3972 Humberto Alencar de Castelo Branco Avenue, 09850-901, SBC, SP, Brazil
<http://www.fei.edu.br/robofei>, amasiero@fei.edu.br

Abstract. This paper presents the description of the RoboFEI@Home team as it stands for the Robocup 2016 in Leipzig, Germany. It describes all mechanical, electrical and software modules, designed for RoboFEI's robot, which is called Judith, a robot based on Peoplebot platform. Judith was built to attempt Human-Robot Interaction (HRI) on social scenarios, which is the main research of our team. A framework for behavior analysis and adaptation have been constructed in the last year to be implement on Judith and many other robots.

Keywords: Robocup@Home, RoboFEI, Peoplebot, Autonomous Robot, Human-Robot Interaction, Service Robotics

1 Introduction

This paper describes the hardware and software aspects of the RoboFEI@Home team, designed to compete in the RoboCup 2016 @Home League.

The first steps towards the first group of research in robots at Centro Universitário FEI was made by Prof. Reinaldo Bianchi in 2002 with the first groups of students in RoboCup Simulation 2D. Next year, with the experience of Prof. Reinaldo Bianchi together with Prof. Flavio Tonidandel, developed the first robots for Very Small Size Category of the IEEE Robotic Competition. In 2004, during the First Brazilian Competition on Intelligent Robots, our team became Brazilian Champion in the IEEE Very Small category. In the same competition, our first team developed for the RoboCup Small Size League became vice-champion. In 2006, Very Small Size team became Champion again, and our first 2D RoboCup Simulation team was the Brazilian champion.

The institution RoboCup Small Size League team, called RoboFEI, won for the first time the Brazilian Robocup in 2010, and is currently the Brazilian champion, winning the championship 6 times in a row (2010 - 2015). This team takes part in the RoboCup World Competition since 2009, and the best result we had was in 2012, when we stayed among the top 8 teams. After developing robotic soccer players for the last 17 years, we developed a team to compete in the RoboCup Humanoid League. The development of this team started in 2012, with students designing and building a humanoid robot from scratch. At

RoboCup 2014, the team stayed among the 16 top teams and in the same year, the team competed the Latin American Robotics Competition (LARC 2014) and won LARC RoboCup Humanoid Kid Size Championship.

Now that the institution has established important positions on small size and humanoid league, we decide to drive our research into new paths for collaborating on social and service robots. The robot design, particularly the human factor concerns, are a key aspect of human-robot interaction (HRI). Research in HRI attractions from similar research in human-computer interaction (HCI) but features a number of significant differences related to the robot's physical real-world personification. This team was born in Usability Engineering Laboratory, now defined as Human Robot Interaction and Intelligent Interfaces Group (HR3iGroup). This research group has abundant knowledge in HCI and is focused on integrating with HRI. The robot's physical personification, simplicity or complexity of design, form and level of anthropomorphism, human behavior and robot interaction feedback, robotic reactions and humanized relationships, are some of the key research areas being explored.

Last year, we applied for and achieved 3rd place at the Latin American @Home competition. We focus on research applied in Human-Robot Interaction, Social and Service Robots, Behavior Analysis, People Modeling and Profiling, and all questions for improving people and robots coexistence.

2 Hardware Design

Our team has one robot, called Judith (see Fig. 1). Judith is a robot based on the Peoplebot platform [1], using a KUKA youBot arm [9] as its main manipulator. As shown in Fig. 1, Judith is composed of: 1 Microsoft Kinect for Xbox 360; 2 sonar arrays; 2 infrared sensors; 1 iPad 2; 1 Yoga HT-320A microphone; 1 touch sensor arrays; 2 speakers; 1 emergency button; and 1 Hokuyo URG-04LX-UG01 laser sensor.

Peoplebot has a vertical grip which was removed for the KUKA youBot arm to be attached. To attach the new arm, we built a support to keep it on the right height to manipulate objects on top of an ordinary sized table. KUKA youBot arm weighs 16.53 pounds so a wooden support was made to hang up with this weight.

Working with two different pre-manufactured robots present some challenges. First of all, each one have electric source with different voltage. KUKA youBot arm needs 24v to work while Peoplebot works with 12v. To put Peoplebot and KUKA youBot arm together, we need to build a circuit for connecting both battery kits together. This circuit is important to keep all parts of the robots safe in case of electric damage on overcharge. It also helps to make both robots work at the same time.

With both mechanical and electrical projects setups, we need to connect both robots on main computer. In order to connect Peoplebot to our laptop we used a serial RS232 to USB converter. The manipulator is able to transmit messages through an EtherCAT cable, which is connected to the computer's network port.

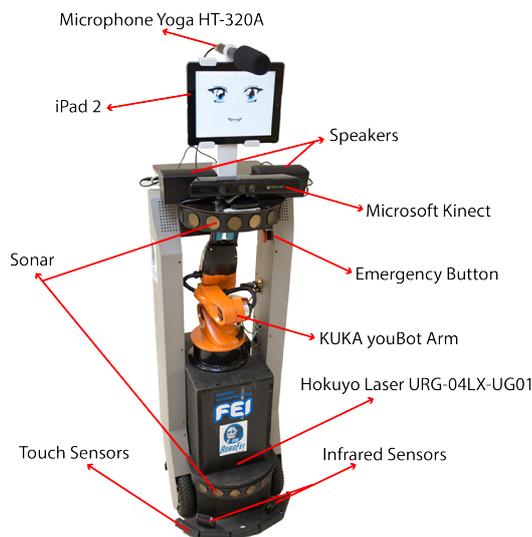


Fig. 1. RoboFEI's Judith

The robot's main computer has been an ASUS Ultrabook 14" Touch-Screen Laptop, with an Intel Core i5, 4GB Memory, 500GB Hard Drive. It has only 3 USB ports supporting all devices, so an USB hub is used to enable more ports and increase the number of interfaced devices. As the iPad needs a paid annual license for software development, a change is being made towards Android technology so as to allow us to develop an interactive face for Judith.

The platform used by Peoplebot makes transportation a challenge, so we have worked on a modular platform using a combination of 3D print technology and aluminum parts. We want to accomplish this project until RoboCup 2016 in Germany.

3 Software Design

This section presents all details about RoboFEI@Home software. A stable version of the software is available on GitHub¹. The section is divided as: (I) Judith's software architecture; (II) face and people recognition; (III) object detection; (IV) speech recognition and synthesis; (V) navigation; (VI) object manipulation.

3.1 Software Architecture

All of Judith's software is developed under Ubuntu Linux 14.04 LTS [22] and ROS Indigo Igloo [18]. The initial architecture of Judith is composed by three layers. Each layer is responsible for one part of the code. The first layer receives

¹ https://github.com/OpenFEI/rfh_judite

all data from sensors, extract the features and publish it for AI and Controller nodes (Second Layer). AI and Controller nodes use the features to run algorithms for computer vision, location, planning, pattern recognition, among others. The second layer then sends to the third layer information regarding the movement of the robot, that is, the velocity of the motors, positions or velocity of the manipulator joints and face expressions to display on the iPad. The third layer controls all the actuators drivers and actions. A graphic view of the architecture is presented on Fig. 2.

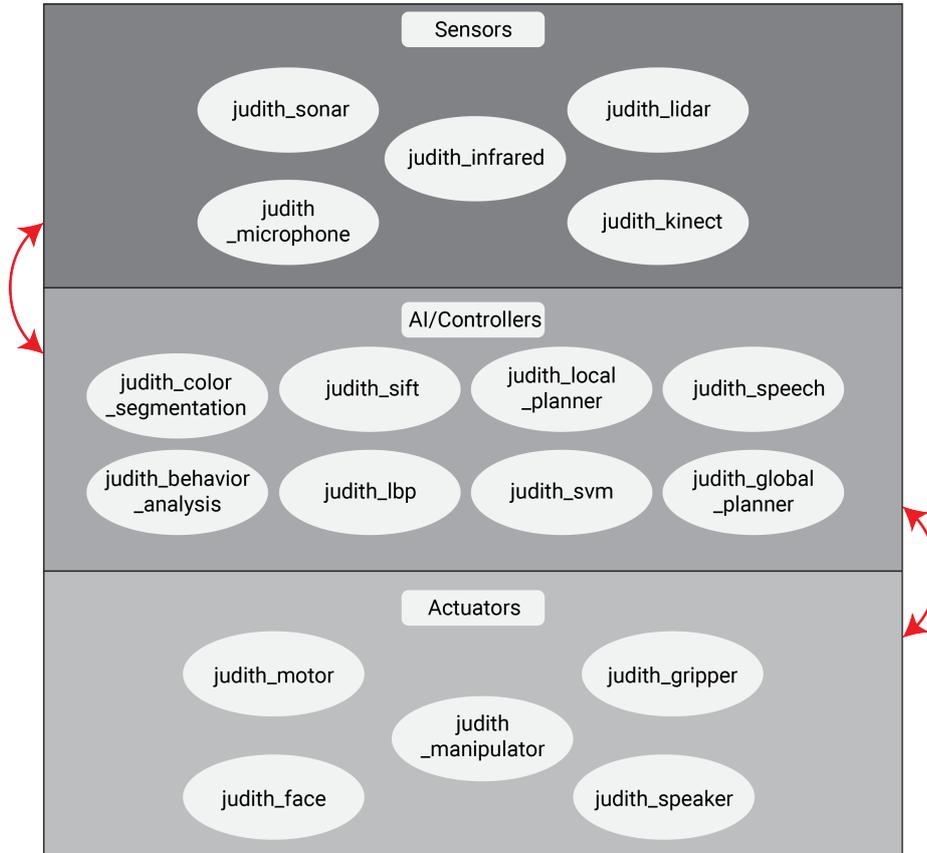


Fig. 2. Software Architecture of Judith

Sensors and actuators layers were developed using C++ [23], due to hardware drivers communication, except by `judith_face` node, which is developed in Java [13] language to create an application on Android [12] Platform. AI/Controllers layer is developed using Python [24] as programming language to all algorithms.

These decisions were made having in mind the best performance of our robot during task execution and also the training new members to the team.

3.2 Face and People Recognition

We understand that, in order to make the user's interaction with the robot more comfortable, it is interesting to recognize and treat them by their name. In that way, we developed nodes that can detect and recognize a person by their face. To make it possible, we acquire an image using a camera (Logitech C920 Webcam or Microsoft Kinect X360 RGB camera) and convert the frame into a OpenCV [6] Image Matrix. This information is used as input for the LBP (Local Binary Pattern) algorithm which has been widely used for face recognition due to its computational performance [2, 25, 21, 26, 20]. LBP algorithm may also help on gender, age and facial expression identification which can be useful when developing a robot's adaptive behavior during interaction.

Beyond facial recognition this set of nodes is responsible for following a single person, without external interference during process. To perform this task, three techniques were combined. First is face detection followed by extraction of the color from person t-shirt [19, 15, 4]. With the t-shirt color defined, we perform color segmentation using OpenCV [14, 17, 7]. The last technique is skeleton extraction through the Nite/OpenNI library for Microsoft Kinect [10]. With this information, the robot can follow the direction of a certain person and also identify the distance between them.

3.3 Object Detection

Object detection is done by using features extraction, detection and matching algorithms. A collection of object textures is kept in the robot's memory as bitmap images, each image being part of a class that represents a particular object. When object detection is called for, the robot extracts features from the stored images using either SURF [5] or SIFT [16] (the algorithm can be chosen during the robot's startup). Then, it extracts features from the camera and matches them with the features from the stored textures. Feature matching is done using brute-force matching: if a certain number of features from a given texture is found in a frame from the camera, then a transformation is done to fit the texture in the frame and a rectangle is drawn around the object, identifying it in the frame.

A more efficient object detection procedure is achieved if the robot is able to extract a background image from the environment. The background can be subtracted from frames in real time and object detection is only performed in parts of the frame that change.

3.4 Speech Recognition and Synthesis

Speech is one of the most important ways to interact with a human. Because of that, we decided to give Judith a female voice, which is generated by the

Festival Speech Synthesis System ², developed by the University of Edinburg. Additionally, voice commands are used to start all of the robots functions. Speech recognition is made using the CMU Pocketsphinx [11].

3.5 Navigation Stack

To interact with the environment, the robot needs to know where it is and how to navigate autonomously. In order to do that, Judith creates a map of the environment using the GMapping application and a Hokuyo URG-04LX-UG01 proximity laser. The map is then saved as a bag using a node from ROS called `map_server`. Map generated by GMapping process is presented on figure 3.

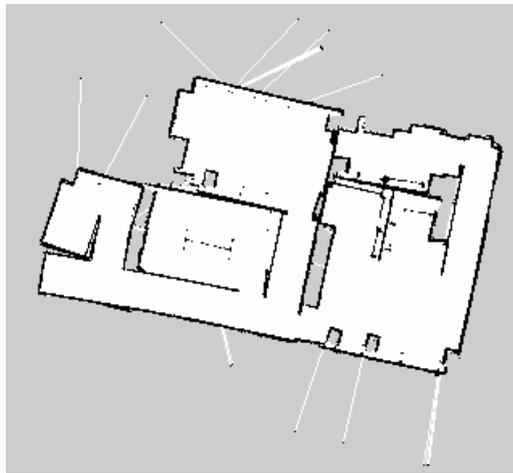


Fig. 3. Map of RoboFEI's Lab

All saved maps can then be used by the robot's simultaneous localization and mapping (SLAM) node. The *Adaptive Monte Carlo Localization* (AMCL) algorithm [8] uses the maps along with data collected from the laser in real time to make the robot capable to locate its position on the map. With the position known, we are able to give the robot a point to go on the map. It then uses the A* algorithm to make the route between the current position and the desired position. During the completion of the path, it executes a *Dynamic Window Approach* (DWA) algorithm to plan movements locally.

3.6 Object Manipulation

For manipulating an object, we first use object detection node (see sec. 3.3) to know its current position. The manipulator executes a series of predetermined

² <http://www.cstr.ed.ac.uk/projects/festival/>

movements to put the gripper on the robot's view. A PCL algorithm [3] is then used to segment the gripper and get object distance. In the final step we execute planning to move KUKA youBot arm joints to get the gripper closer and grab the object.

4 Conclusions

We have described all development of our robot Judith. We believe that all of these tasks can help on home, health and office services. To improve the acceptance of these robots, we have been working into a behavior analysis framework developed using ROS. With this framework, we intend to:

- Identifying people's emotion;
- Creating user behavior profile;
- Improving approach to start a interaction.

With this TDP, we hope to participate on Robocup 2016 in Germany and try to advance during the competition and learn with all teams, as we did to reach third place on XIV Latin American Robotic Competition in 2015.

References

1. Activmedia Robotics, L.: Pioneer 2/peoplebotTM. Operations Manual, Version (2001)
2. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28(12), 2037–2041 (2006)
3. Aldoma, A., Marton, Z.C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R.B., Gedikli, S., Vincze, M.: Point cloud library. *IEEE Robotics & Automation Magazine* 1070(9932/12) (2012)
4. Baggio, D.L.: *Mastering OpenCV with practical computer vision projects*. Packt Publishing Ltd (2012)
5. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (SURF). *Computer vision and image \ldots* 110(September), 346–359 (2008), <http://www.sciencedirect.com/science/article/pii/S1077314207001555>
6. Bradski, G., et al.: The opencv library. *Doctor Dobbs Journal* 25(11), 120–126 (2000)
7. Culjak, I., Abram, D., Pribanic, T., Dzapo, H., Cifrek, M.: A brief introduction to opencv. In: *MIPRO, 2012 Proceedings of the 35th International Convention*. pp. 1725–1730. IEEE (2012)
8. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI 1999*, 343–349 (1999)
9. Group, K.R.: KUKA youBot Robot. <http://www.youbot-store.com/> (2016), [Online; accessed 29-January-2016]
10. Guide, O.U.: Openni organization, november 2010. Last viewed 18, 15 (2011)

11. Huggins-Daines, D., Kumar, M., Chan, A., Black, A.W., Ravishankar, M., Rudnicky, A.I.: Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on. vol. 1, pp. I-I. IEEE (2006)
12. Inc., G.: Google Android. <http://developer.android.com/intl/pt-br/google/index.html> (2016), [Online; accessed 9-February-2016]
13. Joy, B., Steele, G., Gosling, J., Bracha, G.: Java (tm) language specification. Addison-Wesley, June (2000)
14. Kang, S.K., Nam, M.Y., Rhee, P.K.: Color based hand and finger detection technology for user interaction. In: Convergence and Hybrid Information Technology, 2008. ICHIT'08. International Conference on. pp. 229–236. IEEE (2008)
15. Laganière, R.: OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 recipes to master this library of programming functions for real-time computer vision. Packt Publishing Ltd (2011)
16. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. vol. 2, pp. 1150–1157 (1999), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=790410
17. Oliveira, V., Conci, A.: Skin detection using hsv color space. In: H. Pedrini, & J. Marques de Carvalho, Workshops of Sibgrapi. pp. 1–2 (2009)
18. Open Source Robotics Foundation, O.: ROS Indigo Igloo. <http://wiki.ros.org/indigo/Installation> (2015), [Online; accessed 9-February-2016]
19. Pulli, K., Baksheev, A., Korniyakov, K., Eruhimov, V.: Real-time computer vision with opencv. Communications of the ACM 55(6), 61–69 (2012)
20. Samadi, F.: Human age-group estimation based on anfis using the hog and lbp features. Electrical and Electronics Engineering: An International Journal (ELEIJ) 2(1), 21–29 (2013)
21. Shan, C.: Learning local binary patterns for gender classification on real-world face images. Pattern Recognition Letters 33(4), 431–437 (2012)
22. Sobell, M.G.: A practical guide to Ubuntu Linux. Pearson Education (2014)
23. Stroustrup, B.: The C++ programming language. Pearson Education India (1986)
24. VanRossum, G., Drake, F.L.: The Python Language Reference. Python software foundation Amsterdam, Netherlands (2010)
25. Yang, Z., Ai, H.: Demographic classification with local binary patterns. In: Advances in Biometrics, pp. 464–473. Springer (2007)
26. Ylioinas, J., Hadid, A., Pietikainen, M.: Age classification in unconstrained conditions using lbp variants. In: Pattern Recognition (ICPR), 2012 21st International Conference on. pp. 1257–1260. IEEE (2012)

- RoboFEI @Home
- Team Leader: Andrey A. Masiero - email: amasiero@fei.edu.br
- <http://www.fei.edu.br/robofei/>
- Andrey A. Masiero, Douglas R. Meneghetti, Leonardo Contador, Lucas Vasconcelos, Sidney Gitcoff, Flavio Tonidandel and Plinio T. Aquino Junior
- Hardware description:
 - Peoplebot Platform
 - KUKA youBot Arm 5DOF
 - Microsoft Kinect for Xbox 360
 - Front and Rear SONAR Arrays and Front-Facing Upper SONAR Array
 - Hokuyo URG-04LX-UG01 laser
 - Yoga HT-320A microphone
 - Apple iPad 2
- Software description:
 - Ubuntu 14.04
 - ROS Indigo Igloo
 - PCL
 - CMU PocketSphinx
 - SIFT Algorithm
 - LBP Algoritm