Pumas@Home 2020 Team Description Paper [⋆]

Jesus Savage, Reynaldo Martell, Hugo Estrada, Julio Cruz, Marco Negrete,
Diego Cordero, Manuel Pano, Julio Martinez, and Luis Gonzalez

Bio-Robotics Laboratory, School of Engineering
National Autonomous University of Mexico
http://biorobotics.fi-p.unam.mx

Abstract. This paper describes the service robot Justina of team Pumas that has participated in the @Home category of the RoboCup and RoCKIn, both of them international competitions; as well as our latest applied research. These competitions had influenced our architecture in the development of better systems for our service robots by developing algorithms to natural language understanding and the flat-and-textureless object manipulation. In our robotics architecture, the VIrtual and Real roBOt sysTem (VIRBOT), the operation of service robots is divided into several subsystems, each of them has a specific functionality that contributes to the final operation of the robot. By combining symbolic AI with digital signal processing techniques a good performance of a service robot is obtained.

1 Introduction

Service robots are hardware and software systems that assist humans to perform daily tasks in complex environments, to achieve this: they have to be able to understand spoken or gesture commands from humans; to be able to avoid static and dynamic obstacles while navigating in known and unknown environments; to be able to recognize and to manipulate objects and performing several other tasks that a person might request.

Our team has been participated in the category @Home continuously since the start of this competition at the RoboCup in Bremen in 2006. Our team obtained the fourth place and got the award for the best in Speech Recognition and Natural Language Understanding in Nagoya in 2017, in the last years, in the RoboCup 2018 and 2019, the team obtained the second place.

The paper is organized as follows: section 2 enumerates the software components of our robot Justina; section 3 presents overview of the latest research developments in our laboratory; section 4, the conclusions and future work are given; and finally, in Appendix A, the hardware components of our robot Justina.

[⋆] Acknowledgment: This work was supported by PAPIIT-DGAPA UNAM under Grant IG100818

2 Justina's Robotics Architecture

2.1 Software Configuration

Our software configuration is based on the VIRBOT architecture [1], which provides a platform for the design and development of software for general purpose service robots, see figure 1. The VIRBOT architecture is implemented in our robots through several modules that perform well defined tasks [2], with a high level of interaction between them. The principal framework used for interaction is ROS, where a module is represented by one or several ROS's nodes. Also, for modules using the Microsoft operating system, we use our own middleware called Blackboard to link them with ROS nodes running on Linux. In the following subsections are explained each of the layers of the VIRBOT system.

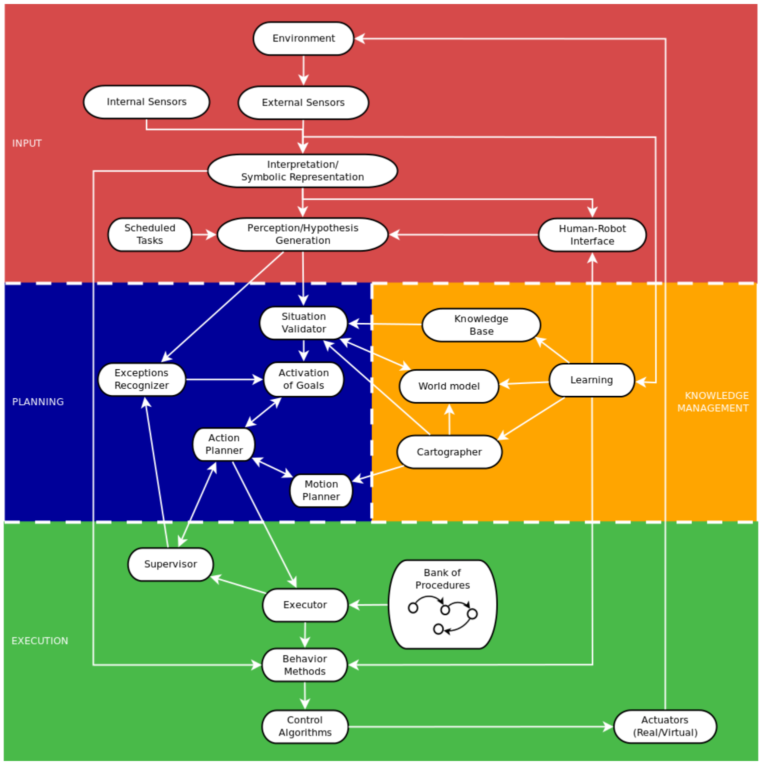


Fig. 1: Block diagram of the ViRBot architecture.

2.2 Inputs Layer

This layer process the data from the robot's internal and external sensors, they provide information of the internal state of the robot, along with the external world where the robot interacts. In some of Justina's designs it has lasers, sonars, infrared, microphones and stereo and RGB-D cameras. Digital signal processing techniques are applied to the data provided by the internal and external sensors to obtain a symbolic representation of the data, furthermore, to recognize and to process voice and visual data. Pattern recognition techniques are used to create models of the objects and the people that interact with the robot. Using the symbolic representation this module generates a set of beliefs, that represent the state of the environment where the robot interacts.

2.3 Planning Layer

The beliefs generated by the perception module are validated by this layer, it uses the Knowledge Management layer to validate them, thus a situation recognition is created. Given a situation recognized, a set of goals are activated in order to solve it. Action planning finds a sequence of physical operations to achieve the activated goals.

2.4 Knowledge Management Layer

This layer has different types of maps for the representation of the environment, they are created using SLAM techniques. Also in this layer there is a localization system, that uses the Kalman filter, to estimate the robot's position and orientation. A rule based system, CLIPS, developed by NASA, is used to represent the robot's knowledge, in which each rule contains the encoded knowledge of an expert.

2.5 Execution Layer

This layer executes the actions and movements plans and it makes sure that they are executed appropriately. A set of hardwired procedures, represented by state machines, are used to partially solve specific problems, such as, person recognition, object manipulation, etc. The action planner uses these bank of procedures and link up some of them that one may generate a plan.

3 Current research

In this section is presented the current research developed in our laboratory to improve the performance of our service robots.

3.1 Natural language understanding

Natural language understanding is used in order to the service robot interprets the language and then perform an especific task. One of the main problems using natural language understanding is the representation of meaning. We have a framework for defining the semantics. The robot's semantics are therefore instructions that allow it to carry out relevant operations.

Conceptual Dependency (CD) is a theory, developed by Schank [3], for representing the meaning contained in sentences. This technique finds the structure and the meaning of a sentence in just one step. It is useful to represent sentences using this technique when there is not a strict grammar associated with the sentences, and also when the objective is to make inferences from them. The CD representation of a sentence is built using conceptual primitives, these represent thoughts and the relationships between thoughts. Using conceptual dependency facilitates the use of inference rules, because many inferences are already contained in the representation itself. There are several primitives to represent actions, for example two of the more commonly used are the following:

ATRANS: Transfer of an abstract relationship (e.g., give.)

PTRANS: Transfer of the physical location of an object (e.g., go.)

Each primitive represents several verbs which have similar meaning. For instance give, buy, steal, and take have the same meaning, i.e., the transference of one object from one entity to another one. Each primitive is represented by a set of rules and data structures. Basically each primitive contains the following components:

An Actor: He is the one that perform the ACT.

An ACT: Performed by the actor, done to an object.

An Object: The action is performed on it.

A Direction: The location that an ACT is directed towards.

A State: The state that an object is in, and is represented using a knowledge base representation as facts in an expert system.

For instance the phrase: **"Robot, please give this book to Mary"**, when the verb give is found in the sentence an ATRANS structure is issued.

(ATRANS (ACTOR NIL) (OBJECT NIL) (FROM NIL) (TO NIL))

The empty slots (NIL) need to be filled finding the missing elements in the sentence. The actor is the robot, the object is the book, etc, and it is represented by the following CD:

(ATRANS (ACTOR Robot) (OBJECT book) (FROM book's owner) (TO Mary))

CDs can be use for representing simple actions. It is also well suited for representing commands or simple questions, but it is not very useful for representing complex sentences. The CD technique were implemented in an expert system.

Much of the human problem solving or cognition can be expressed by IF THEN type production rules. Each rule corresponds to a modular collection of knowledge call chunk. The chunks are organized in loose arrangement with links to related chunk of knowledge, reasoning could be done using rules. Each rule is formed by a left side that needs to be satisfied (Facts) and by a right side that produce the appropriate response (Actions).

IF Facts THEN Actions.

When an action is issued by a rule it may become a fact for other rules, creating links to other rules. A system may use thousands of rules to solve a problem, thus it is necessary a special mechanism that will select which rules will be fired according to the presented facts. That mechanism is an Expert System "Engine". The Inference Engine makes inferences by deciding which rules are satisfied by facts, prioritize the satisfied rules, and executes the rule with the highest priority. This expert system provides a cohesive tool for handling a wide variety of knowledge with support for three different programming paradigms: rule-based, object-oriented, and procedural. The data of the humans interacting with the robot, of the objects and the locations is represented using facts that contain several slots with information related with them. The Robot is able to perform operations like grasping an object, moving itself from on place to another, finding humans, etc. Then the objective of action planning is to find a sequence of physical operations to achieve the desired goal. These operations are represented by a state-space graph.

In the previous example, when the user says **"Robot, please give this book to Mary"**:

> (ATRANS (ACTOR Robot) (OBJECT book) (FROM book's owner) (TO Mary))

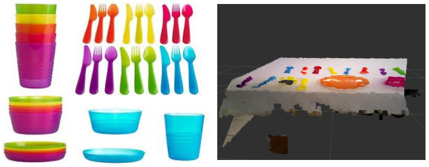
All the information required for the actions planner to perform its operation is contained in the CD and knowledge data base. Our system has been described in [4] and successfully tested in robotics competitions [5], as the RoboCup and RockIn [6], in the category @Home. In RoboCup@Home 2017 our robot was awarded as the best in Speech and Natural Language Understanding.

3.2 Intelligent flat-and-textureless object manipulation in Service Robots

Grasping objects from a table is a common task for Service Robots. Robots mounted with sensors suach as RGB-D cameras are able to detect the dominant plane, cluster point clouds on the plane and analyze them in order to detect the objects on it and determine the best strategy to manipulate the objects as in [7] and [8]. However, when the objects lack texture and or volume (i.e. flat objects), the manipulation task becomes more challenging.

In the 2018 edition of Robocup@Home, one of the tests was the Procter & Gamble (TM) Dishwasher Challenge (PGDC)in which they have to remove all objects from a table and place them into a dishwasher [9]. The set of objects that were used during the test are made of plastic, without any visual pattern on them but colours with high contrast, as shown in Figure 2a. Besides, the objects' surface is very reflective provoking a significant variation in colour. In consequence, the information sensed by the RGB-D camera results noisy and the objects seem to belong to the table, as shown in Figure 2b.

We present our solution to the flat and textureless object detection and recognition problem. We first use colour features in the detection process and then extract the geometry properties using the point cloud to describe them. This information also results useful in the manipulation process.

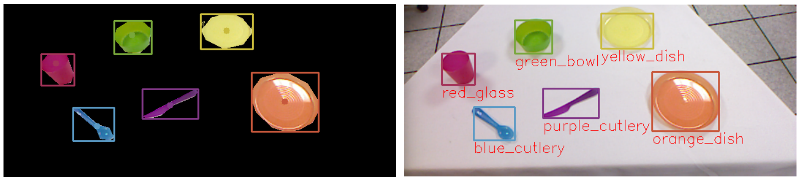


(a) (b)

Fig. 2: a) Tableware and cutlery objects used in the RoboCup@Home's P&G Challenge and b) table setup 3D view from an RGBD camera. It can be observed that the reconstructed scene presents some errors due to sensor noise, varying illumination conditions and high object reflectance.

For background subtraction, we find the range of values in each channel on the HSV colour space for all objects and a raw segmentation mask is obtained where morphological operations of closing and dilation are applied to close the gaps where segmentation was wrong. To deal with cases where objects present significant illumination changes, a convex hull operation is applied as the final step in the segmentation process. Finally, we obtain an RGB mask and a 2D bounding box per object.

We then categorise objects into four different classes, namely glass, dish, bowl, and cutlery, by normalising the image with respect to the average depth and then extracting the area in the RGB image and the number of points and the eigenvectors from the point cloud. An example of the classification result is shown in Figure 3.



(a) (b)

Fig. 3: a) Typical output after object detection using colour features b) Classification result using 3D information to a setup given.

In our implementation, after the objects on the table are detected and recognised, this information is sent through ROS messages to the action planner module. The message contains information about each object, such as: a) related to the point cloud: eigenvalues, eigenvector, size, nearest point and centre point in robot coordinates, b) related to the image: position of top left corner, width and height of the bounding box, c) related to the object: centroid, roll, pitch and yaw manipulation angles, and a priority constant (this variable is set given the grasping ability of a given object with the robot hand – the priority, in increasing order, is as follows: dish, cutlery, bowl and glass).

Due to the size and shape of the objects in this test, we use a feedback strategy based on the visual information to verify whether the object was taken.

In the PGDC test, robot "Justina" obtained a good performance in the manipulation tasks, as we can describe in [10] where the robot was able to grasp 3 out of 4 objects that the robot attempted to manipulate, two of them in the first attempt, and the other in the second attempt after visual feedback confirmation was received.

4 Conclusions and future work

It is clear, that during the 11 years in which our team Pumas has been participated in the RoboCup and 2 years in the Rockin [6] in the category @Home, the performance and research developed, in the service robot area, in our laboratory has been improved considerably. Our service robot architecture, the VIRBOT, has been evolving according to the requirements that these robotics competitions asked each year. In these years, the full system has been improved, both in hardware and software, having reliable performance and showing promising results. Particularly, this year, we add an Embedded Develotment kit [11] for perform GPU's solutions. In terms of software, we have change the way of conceiving the tests of the competition: passing from static state machines to inferred action planning generated by a rule based system. As for future work, we are working in a new configuration for the omnidirectional base to improve the navigation performance. Also, we will explore new topics as the memory and enviromental reasoning of the robot, in order to explain things that hapenned in the past.

References

1. *ViRbot: A System for the Operation of Mobile Robots*, Savage, Jesus and et al, RoboCup 2007: Robot Soccer World Cup XI, pp 512-519, Springer Berlin Heidelberg, 2007.
2. *The Design of Intelligent Agents: A Layered Approach*, Muller, Jorg P,Springer-Verlag New York, Inc.1997.
3. *Conceptual dependency and its descendants*, Steven L. Lytinen, Computers & Mathematics with Applications, 1992.
4. *The Use of Expert Systems for Semantic Reasoning in Service Robots*, Jesus Savage, Julio Cruz, Reynaldo Martell, Hugo Leon, Marco Negrete, and Jesus Cruz, 2nd Workshop on Semantic Policy and Action Representations for Autonomous Robots (SPAR), IROS 2017
5. *The Role of Robotics Competitions for the Development of Service Robots*, Jesus Savage, Marco Negrete, Mauricio Matamoros, Jesus Cruz, IJCAI'16, Workshop on Autonomous Mobile Service Robots, New York, USA, 2016.
6. *RoboCup@Home* http://www.robocupathome.org
   Rockin http://rockinrobotchallenge.eu/home.php

7. *Grasping novel objects with depth segmentation*, D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng, 2010 IEEE/RSJInternational Conference on Intelligent Robots and Systems, pp. 2578–2585, Oct 2010.

8. *Object detection based on plane segmentation and features matching for a service robot*, A. J. R. Neves, R. Garcia, P. Dias, and A. Trifan, International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 10, no. 4, pp. 775 – 782, 2016.

9. *Robocup@home 2018: Rules and regulations*, M. Matamoros, C. Rascon, J. Hart, D. Holz, and L. van Beek, 2018

10. *Intelligent flat-and-textureless object manipulation in Service Robots*, A. Ortega, H. Estrada, E. Vázquez, R. Martell, J. Hernández, J. Cruz, E. Silva, J. Savage, and L. Contreras, IROS 2018 Workshop ”Towards Robots that Exhibit Manipulation Intelligence”, 2018.

11. *Jetson* https://developer.nvidia.com/embedded/buy/jetson-tx2-devkit

5 Team Information

Name of Team:

Pumas

Contact Information:

Jesus Savage

Bio-Robotics Laboratory

School of Engineering

National Autonomous University of Mexico

robotssavage@gmail.com

Web Site:

http://biorobotics.fi-p.unam.mx

Team Members:

Jesus Savage, Reynaldo Martell, Hugo Estrada, Julio Cruz, Marco Negrete,

Jaime Marquez, Manuel Pano, Julio Martinez, Jesus Cruz, Jose Cruz

Description of Hardware:

Justina’s Robotics Architecture (cf. section 2)

Description of Software:

Most of our software and configurations are open-source and can found at:

https://github.com/RobotJustina/JUSTINA

Operating System	Ubuntu 16.04 LTS; Windows 7 VM
Middleware	ROS Kinetic; Blackboard
SLAM	ROS Gmapping
Navigation	Navigation using Kinect + Ocupancy grid + A*
Object Recognition	Histogram Disparity + YOLO
Face Detection	Haar Cascades
People Detection	YOLO
Gesture Recognition	OpenPoses
Face Recognition	Facenet
Speech Synthesis	Loquendo
Speech Recognition	Microsoft Speech Recognition
Inference Engine	CLIPS

A Appendix

A.1 Hardware Configuration

Our service robot Justina, see figure 4, has the following components:

HARDWARE:

- **Mobile base:** Omnidirectional through differential pair configuration and omnidirectional wheels.
- **Manipulators:** 2 x 7-DOF anthropomorphic arms with 10 Dynamixel servomotors each.
- **Head:** 2-DOF (Pan and tilt) built with Dynamixel servomotors.
- **Torso:** 1-DOF (Elevation) through a worm screw and a configuration of gears.
- **Speakers:** Two speakers to generate synthetic speech.
- **RGB-D Camera:** Microsoft's Kinect sensor.
- **RGB Camera:** Logitech Pro C920 Full HD.
- **Microphone:** Rode NTG2 directional microphone.
- **Array of Microphones:** An array of four microphones to detect sound sources.
- **Laser:** Hokuyo rangefinder URG-04LX-UG0.

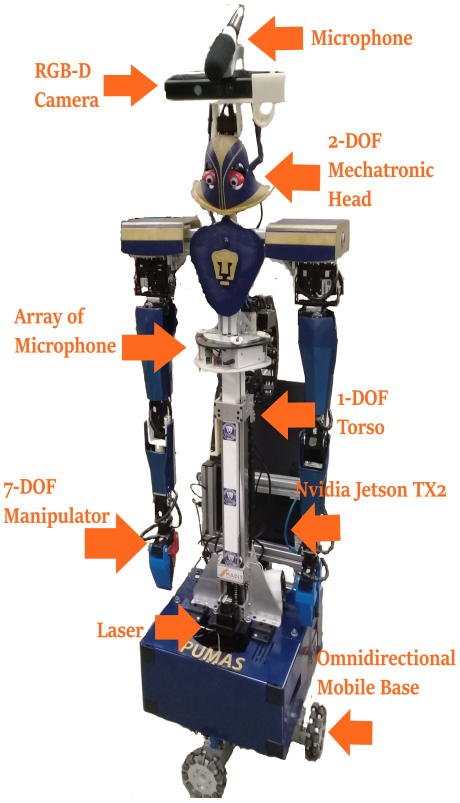


Fig. 4: Robot Justina

- **Embedded System:** NVIDIA Jetson TX2 to image processing.