Pumas@Home 2014 Team Description Paper

Bio-Robotics Laboratory, Universidad Nacional Autónoma de México, UNAM
`http://biorobotica.fi-p.unam.mx/`

Abstract. This paper describes an autonomous service robot called Justina, built by the team Pumas in the Bio-Robotics Laboratory at the National University of Mexico. This robot is based on the ViRbot architecture for autonomous mobile robot operation, which defines a human-robot interaction interface made of three main layers: Input layer, Planning and Knowledge Management layer, and Execution layer. The Input layer includes all algorithms related to the acquisition of data from the environment. The Planning and Knowledge Management layer performs most of the AI algorithms and the Execution layer includes low-level control and supervision. The implementation of the ViRbot is made through several modules (executables) that perform well defined tasks and have a high interaction. The information exchange between modules is made through a central module called Blackboard, which supports shared variables, with publisher/subscriber pattern, and message passing.

1 Introduction

The service robots are hardware and software systems that can assist humans to perform daily tasks in complex environments. To achieve this, a service robot has to be capable of understanding commands from humans, avoiding static and dynamic obstacles while navigating in known and unknown environments, recognizing and manipulating objects and performing other several tasks that the human beings ask for.

The main objective of the ViRbot architecture [1], is to operate autonomous robots that can carry out daily service jobs in houses, offices and factories. This system has been tested in the last six years in the RoboCup competition at the @Home category and will be used again at the competition in João Pessoa, Brazil in 2014. ViRbot is implemented in the robot Justina (see Figure 1) and integrates the work of several

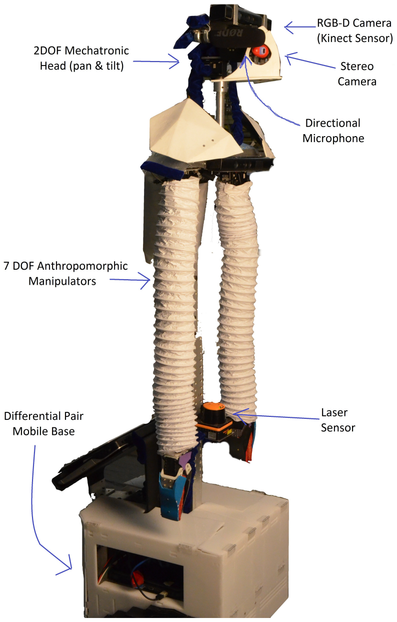


Fig. 1: The robot Justina

research areas such as expert systems, computer vision, task, path and motion planning, machine learning and automatic control.

This paper is organized as follows: section 2 is an overview of the ViRbot Architecture; this system provides a platform for the design and development of robot Justina's software. Section 3 describes the implementation of ViRbot architecture as a set of software modules that perform different types of tasks such as the hardware control, task planning, sensor data acquisition and a Blackboard for the communication management. In Section 4 conclusions and future work are given.

2 ViRbot: A System for the Operation of Mobile Robots

ViRbot architecture divides the operation of a mobile robot into several subsystems, each of which can be classified in one of three main layers: Input, Planning and Knowledge Management, and Execution. Figure 2 shows a schematic diagram of the ViRbot subsystems and the interactions between them.

2.1 The Input Layer

Interpretation/Symbolic representation. This subsystem performs the extraction of high level characteristics of the raw sensor information. The goal of this subsystem is to obtain a symbolic representation of the world that will be used for hypothesis generation.

Perception/Hypothesis Generation. This subsystem's task is to generate hypothesis from the information given by the Interpretation module. The hypothesis generation is performed taking into account a set of *Scheduled Tasks* and the information given by the Human-Robot Interface.

Human/Robot Interface. The purpose of this subsystem is to recognize and process the voice and gesture commands. It is divided in three modules. The first one is the *Speech Recognition* module. It uses digital processing algorithms to analyze the voice commands given to the robot. The *Natural Language Understanding* module, which is the second one, is used to find a symbolic representation of spoken commands given to the robot using the recognized sentences coupled with Conceptual Dependency techniques. Finally, the *Conceptual Dependency* module [2] is used to represent meaning by finding the structure and meaning of a sentence in a single step. It allows rule based systems to make inferences from a natural language system in the same way humans do, using conceptual primitives that represent thoughts, actions, and the relationships between them.

2.2 The Planning and Knowledge Management Layer

Situation Validator. The goal of this subsystem is to validate the hypothesis generated by the Hypothesis Generation module. It takes the information of the Knowledge Base, World Model and Cartographer and validates the hypothesis.

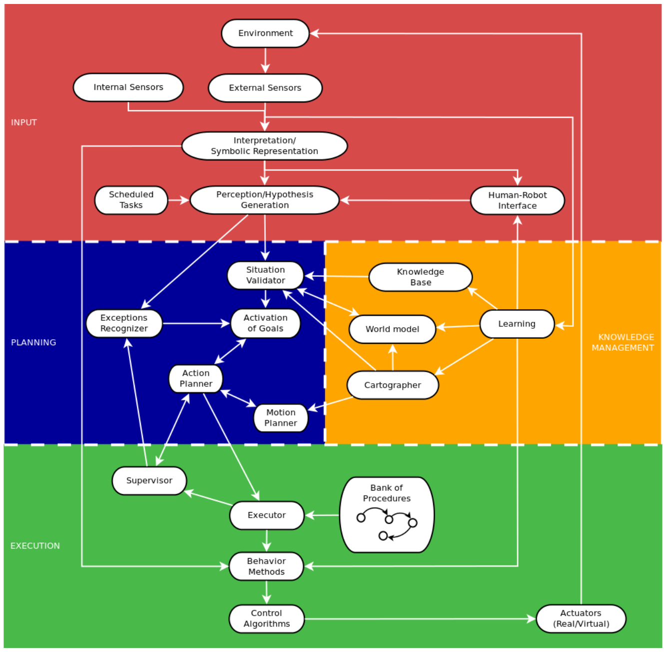


Fig. 2: The ViRbot Architecture

Once a set of hypothesis are validated, a situation is generated and a set of goals are activated.

Activation of Goals. When a situation is generated by the Situation Validator subsystem, a set of global and local goals are activated in order to solve such situation.

Knowledge Base. This subsystem contains the procedural knowledge of the robot. This knowledge is stored as a set of if-then pairs that constitutes a rule-based expert system.

World Model. This subsystem constitutes the declarative knowledge of the robot. It is composed by maps of the environment, high-level characteristics of the objects the robot is capable to recognize and a data base of the faces the robot can identify.

Cartographer. It maintains the robot spatially localized. This module uses *Raw Maps*, *Topological Maps* and *Symbolic Maps* to represent the environment and Hidden Markov Models [3] and Kalman Filters to estimate the robot position.

Learning. The following learning algorithms are used for the robot:

1. *Map building.*- Cluster sets are used to locate and represent obstacles and free space in the environment.

2. *Self Localization.*- Hidden Markov Models using Vector Quantization & Self Associative Neural Networks are used to estimate the robot's position and orientation.
3. *Behaviors.*- Genetic Algorithms are used to learn new behaviors.

Action Planner. The objective of the action planning is to find a sequence of physical operations to achieve the desired goal. This module takes as input the output of the Activation of Goals module and has a high interaction with the Motion Planner and with the Execution layer in general.

Motion Planner. This subsystem performs path planning (global goal) and obstacle avoidance (local goals) to take the robot from one point to another. It takes the information coming from the Action Planner to establish a global goal and uses the Cartographer to localize the robot.

Exceptions recognizer. This module has the capability to detect errors in the task execution. According to the outputs of the Hypothesis Generator and the Supervisor, which will be described in the following subsection, it modifies the set of activated goals and thus the plan to solve the current situation is also modified.

2.3 The Execution Layer

Bank of Procedures. It is a set of hardwired procedures used to solve tasks with a low level of complexity or that require a simple and well defined routine.

Executor. It manages the execution of tasks according to the output generated by the Action Planner. This module uses the information of the Bank of Procedures to execute a complex task as a set of several simpler tasks.

Behavior Methods. A set of solutions encoded in State Machines, which are used in repetitive tasks. For example, in obstacle avoidance, the repetitive task consists of following a sequence of goals until reaching the final one.

Control Algorithms. Control algorithms, like PID and nonlinear techniques, are used to control the actuators of the robot (head, arms and mobile base). The set point or desired value for the actuator is given by the Behavior Methods subsystem.

Supervisor. This module is similar to the Exceptions Recognizer but in a lower level. Actually, the output of the Supervisor is one of the exceptions that can be recognized.

3 ViRbot implementation on Robot Justina

ViRbot is implemented in robot Justina by means of a Blackboard architecture. This allows to use more than one platform and run software modules programmed in different languages in different computers. Robot Justina uses computers running both Linux and Windows, and modules programmed in C#, C++, Python and CLIPS.

3.1 Software

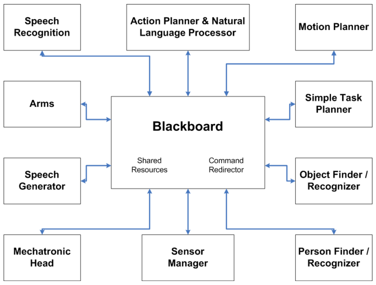


Fig. 3: Blackboard structure

Blackboard. This is a flexible system to control the transmission of messages between modules, monitor their status and store the common data used by all the modules that integrate the system. All common data is stored in the Blackboard as shared variables to grant access at any time to other modules. It also uses a producer/subscriber paradigm to enhance the Real Time Awareness of the system and reduce communication overloads. Also the Blackboard offers a flexible platform to implement ViRbot as shown in figure 3.

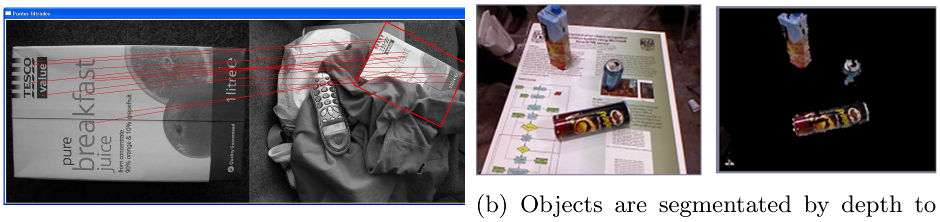
Action Planner. For the Action Planner, the rule-based expert system CLIPS, developed by NASA [4], is used. With CLIPS it is easy to implement tasks of several subsystems of ViRbot Architecture such as the Robot's Tasks, Perception and Knowledge Representation subsystems. The Action Planner works in the highest level of abstraction, coordinating the tasks that the robot must perform and choosing the adequate behavior in each situation.

Simple Task Planner. In the ViRbot system, the Bank of Procedures involves simple and repetitive tasks which require low level of planning and that can be easily achieved by state machines (like look for a person/object or grasp an object). Those tasks are performed by the Simple Task Planner module. Also this module incorporates reactive behaviors if the Action Planner is not available.

Motion Planner The motion planner is responsible for finding the best sequence of movements to reach the final destination given by the Action Planner or the Simple Task Planner combining classic and modern techniques. It uses Cartographer's maps to calculate the path planning. In parallel to the geometrical representation, it generates a topological representation of the environment and, by Dijkstra algorithm [5], finds the optimal path between the current and goal positions. Obstacle avoidance is achieved using Potential Fields, whose gains are tuned by a genetic algorithm, and Finite State Machine based behaviors. Localization is performed by a Kalman Filter and coordinated movements of the mobile base and mechatronic head.

Vision Subsystem. The Vision subsystem has been specialized in three modules: Object Finder, Person Finder and Skeleton Finder.

Object Finder. The Object Finder module features a robust implementation of an object tracker where objects are segmented using receptive fields [6][7][8], and represented by feature points which are described in a multi-resolution framework, that gives a representation of the points in different scales.



(a) Visual identification of objects

(b) Objects are segmentated by depth to reduce compute time and enhance accuracy

Fig. 4: Object recognition and location using depth sensors

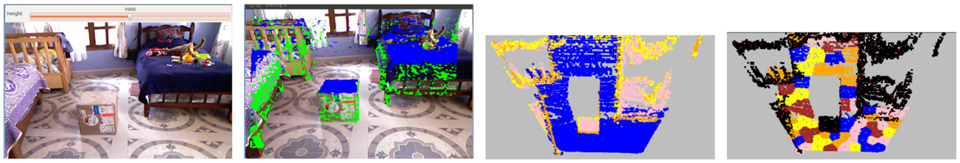
Detection and description of interest points are based on the SIFT (Scale-Invariant Feature Transform) algorithm [9] after a first segmentation by depth is made (see figure 4a). After an object is recognized, the geometry of the object is computed from the depth map to fetch the object's centroid and orientation[10].

This module can also extract geometric characteristics of the environment such as planes, lines, corners and spikes.

Person Finder. The Person Finder Module uses VeriLook SDK for multiple face detection and recognition. The name associated to the detected faces, if known, and it's confidence are stored in the Blackboard.

Skeleton Finder. This module uses the Microsoft Kinect SDK to identify persons and it provides the position of the humans near to the robot.

Cartographer. The Cartographer module stores the several maps used by the motion planer and also generates topological maps with the 3D points obtained from the depth map of the Kinect module.



(a) Environment (b) Plane detection (c) Navigable area (d) Topological map

Fig. 5: Topological map generation from 3D points.

Speech Recognition. This module is part of the Human/Robot Interface ViRbot subsystem and uses the MS SAPI 5.3. An array of hypothesized strings with its confidences is stored in the Blackboard to let the Action Planner to choose the most adequate candidate for the robot's current context.

Speech Generation. This module is part of the ViRbot's Human/Robot Interface and allows the robot to interact with humans by synthesizing voice. It uses the Microsoft SAPI 5.3 with the Loquendo Susan voice.

Sensor data processing. The Sensors Data Processing module controls the data acquisition from several sensors such as lasers and sonars. The acquired data are stored in the Blackboard to be used by other modules.

3.2 Hardware

Differential Pair Mobile Base. The robot uses a differential pair mobile base to navigate. A microcontroller-based board sends and receives commands for motor control and encoder reading, using a serial communication interface.

Manipulator. The robot has two 7 DOF articulated arms with anthropomorphic configuration built with Dynamixel servomotors and a microcontroller based embedded system for control and path planning. The arm has an anthropomorphic design in order to perform a better obstacle avoidance and natural human-like movements. For path tracking, a PID plus gravity compensator control is used and a vision based control is implemented for handling objects.

Sensors. The robot has several sensors for getting information on the surrounding environment: laser range finder for motion planning and obstacle avoidance, a Kinect system for seeking humans and objects, a stereo VGA camera for pattern recognition and a directional microphone for natural-speech command interpretation. Digital signal processing techniques are applied to the obtained signals to interact with the dynamic environment.

Mechatronic Head. The Mechatronic head design is based on the corresponding movements of the human head with 2 degrees of freedom: pan and tilt. This freedom of movement allows to point the sensors to obtain accurate readings of the environment and perform a systematic search in a particular zone of interest. It carries three different sensing devices on it: a Kinect system, a directional microphone and a stereo camera. The sensors are integrated with a friendly plastic face providing confidence to humans that interact with the robot.

4 Conclusions

The ViRbot system was successfully tested in the Robocup@Home category in the last four RoboCup competitions and in Atlanta 2007, the robot obtained the third place in this category. In these years, the full system has been improved having reliable performance and showing promising results. As future work, the structure of the robot will be redesigned to reduce its weight and make its assembly and maintenance easier. Also, new algorithms for the recognition of human faces and objects, along with localization, 3D mapping and more complex behaviors arbitrated by the Action Planner's expert system.

To improve navigation, SLAM technics are being developed using the visual relationship between two different views of the same scene. Also, to provide compatibility with the most used framework for robots, a bridge between Blackboard and ROS is being developed.

Some videos showing the operation of the ViRbot system can be seen at `http://biorobotics.fi-p.unam.mx` .

References

1. Jesús Savage, Alfredo Weitzenfeld, Francisco Ayala, and Sergio Cuellar. The use of scripts based on conceptual dependency primitives for the operation of service mobile robots. RoboCup 2008 Symposium, Suzhou, China, 2008.
2. Roger C. Schank. *Conceptual Information Processing.* North-Holland Publishing Company, 1975.
3. S. Thrun and A. Bücken. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the AAAI Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, 1996.
4. Gary Riley. *CLIPS Reference Manual Version 6.0. Technical Report Number JSC-25012.* Software Technology Branch, Lyndon B. Johnson Space Center, Houston, TX, 1994.
5. S. M. LaValle. *Planning Algorithms.* Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.
6. S. Ekvall and D. Kragic. Receptive field cooccurrence histograms for object detection. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 84 – 89, 2005.
7. Tse-Wei Chen, Yi-Ling Chen, and Shao-Yi Chien. Fast image segmentation and texture feature extraction for image retrieval. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 854 – 861, 272009-oct.4 2009.
8. Peng Chang and John Krumm. Object recognition with color cooccurrence histogram. In *Proceedings of CVPR '99*, 1999.
9. David G. Lowe. Distinctive image features from scale-invariant keypoints, 2003.
10. José Figueroa, Luis Contreras, Abel Pacheco, and Jesus Savage. Development of an object recognition and location system using the microsoft kinecttm sensor. In *RoboCup*, pages 440–449, 2011.