Pumas Team Description Paper 2023

Mitzi Ramírez, Rebeca Muñoz, Jesús Savage, Marco Negrete, Julio Martínez, Daniel Ruelas, Miguel Sánchez, Isaac Hernández, Joshua Martínez, and Saul de la Rosa

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

Abstract. This paper describes the development made for the robot Festino of team Pumas. It is intended to participate in the 2023 Mexican Tournament of Robotics and the 2023 RoboCup Logistics League. The work done with the robot includes Navigation Planning, Image Processing, Use of the Point Cloud, Pattern Recognition, Environment Representations, and Action Planning, among others. We expect to transfer the VIrtual and Real roBOt sysTem (VIRBOT) to do the Action Planning, combined with the Analysis of the Environment to obtain a composed behavior of the robot that completes the tasks required in the Logistics League.

Keywords: Image Processing, RGB-D camera, Grasping, Navigation, Object Detection, RoboCup, Logistics League, Action Planning.

1 Introduction

The Pumas team has its origin in the RoboCup competition in the OPL and DSPL categories of the @Home League with the robots Justina and Takeshi (HSRB by Toyota), respectively, being able on several occasions to be in the Top 3 winners of the competitions. The team begins as part of the Bio-Robotics Laboratory of the School of Engineering of the National Autonomous University of Mexico.

In 2023, for the first time, the Pumas team is looking to compete in the Logistics League. Considering the obvious differences between the two Leagues we think some of the development done for the previous robots can be translated into the Logistics League, as can be the navigation and some parts of the system used in the decision-making for robot Justina, nonetheless, the work for this category is being made currently and we would like for it to be tested for the first time in real competition-like conditions on the 2023 RoboCup and set the basis of the work in this League in the future of the Bio-Robotics laboratory.

2 RoboCup Logistics League

The accelerated advance of the industry leads to giving the robots the ability to sense their environment and plan the most efficient way to complete a task, as well as including new and more powerful equipment to follow the rhythm of the industry. The Logistics League aims to simulate an industrial-like environment, in which teams can demonstrate the development made for such tasks.

In this specific setting, the robot is supposed to take initial pieces and move them to different stations where they are going to be processed to obtain a final-composed piece. Each one of these stations (MPS) does a different part of the simulated production line and the robot must help with this task moving sequentially the pieces from one station to another to complete the process.

To be capable of such activities the robot must move around the workspace and interact with each MPS. It must be able to avoid mobile and static obstacles in its way and localize itself in the space. Once the navigation part is assured, the next task would be to integrate an algorithm that makes the robot navigate freely, explore its environment, and find representative marks to identify the objects that surround them.

Added to these conditions the robot must be able to grasp the pieces, this makes it useful to implement an active-vision system with which the robot will be able to see, identify, and classify the object and then grasp it. On the top level of these activities, a central system must know the state of the production process and the robot, as well as the sequence of actions that compose the whole line. For this, it is necessary to use a system to generate a plan and then integrate the new information to it to be able to react to the change of conditions of the environment.

Reactive behaviors to different mistakes are important, such as not positioning the pieces correctly in the conveyor belt and them falling off of it. The robot should notice this situation, stop the process, and then re-plan the strategy to restart or discard the process for that order.

For all of these tasks it is imperative to listen to instructions given by a central system. The software that does this management is called *Referee Box (RefBox)*, it is in charge of evaluating, monitoring and controlling the game, it is also the communication bridge between the referees and the participating teams. The RefBox gives to the teams the information of the stage of the competition, the state of the MPS, deliveries of the orders, among other important information. To communicate with the teams it uses User Datagram Protocol (UDP) and we must be constantly receiving the information, given the conditions of the competition, where the internet connection is not always available or it is not always stable, the messages are published continuously to avoid information loss [5].

In previous editions of the competition some teams have a central system that reunites the information of the robots, stations and then assign the tasks according to a determined plan to optimize the time and number of steps that need to be done to accomplish the goal in time [1].

3 Team Summary

The Pumas team is newly formed. It is integrated by a combination of electrical-electronic, mechatronic, and computer engineers and students, as well as computer scientists. Our collaborators are working to obtain their bachelor's, master or Ph.D. degrees. Some of them have experience working on Justina or Takeshi robots as well as other categories of the RoboCup, and some others are completely new to the competition but have experience in robotic-related topics. We are under the guidance of Dr. Jesus Savage and Dr. Marco Negrete, both of them are used to the RoboCup competition and are experienced engineers and researchers.

4 Festino's Architecture

Being the first time participating in this category our focus is still on the development of the low-level elements of what is going to be our future architecture. The main requirement for the robot is to use the Robotino base, then be able to add the peripheral systems freely. In this case, we have integrated an RGB-D camera, a gripper that includes a magnetic proximity sensor, and an infrared sensor, in addition to the LiDAR laser that is included in the robot. Normally we mount the computer that is going to do the processing, connected to the robot with an Ethernet cable or through WiFi connection.

5 Festino Robot

Fig. 1: Gripper

6 Software

6.1 Connection with RefBox

The team communication is made using protocol buffers by Google to obtain the information and then send these messages to the planning stage, which is going to translate this information into commands to be used in our ROS system to control the robot [2].

6.2 Planning

Our software configuration so far is based on State Machines, given the condition that we only have one robot, the planning scheme is very simple, and the actions are predefined as to GoThere-DoThat. We are currently working on implementing our architecture with the VIRBOT [4] structure.

The VIRBOT architecture is separated into four different layers, the **INPUT** that the robot receives from the environment, the **PLANNING** that is made with CLIPS, and the **EXECUTION** that we make with ROS. In this system, ROS is used to communicate the low-level parts of the system, our actuators, cameras, and different sensors that compose the robot, all of them are communicated using ROS nodes.

Input This stage is composed of the information that the robot receives from the sensors and they define the robot's perception of the environment and his own internal state. Some of the sensors are the RGB-D camera, and the LiDAR laser scanner, with the information that the robot gets from the sensors it forms the environment description and is able to interact with its surroundings.

Planning The assumptions that are being made by the perception module are validated here, using Knowledge Management and creating a situation recognition which activates a set of goals that ideally are going to find a solution, establishing a set of tasks for the robot to finish a task.

Knowledge Management In this stage, an expert system is implemented, in which a set of maps are created using a combination of SLAM techniques and a localization system that uses a Kalman filter to generate an environment description that is going to be processed by CLIPS to generate a set of rules to give the robot the definition of its knowledge.

Execution This is the stage where the plans and the different tasks that compose them are executed by the robot, this specific tasks translated to actions could be FindObject, GotoPlace, AlignMPS, AlignObject, etc. Defined as individual actions that give the robot the capacity to complete complex instructions.

We are also working on developing a Simulator that allows us to practice the planning task without the limitation of only having one robot.

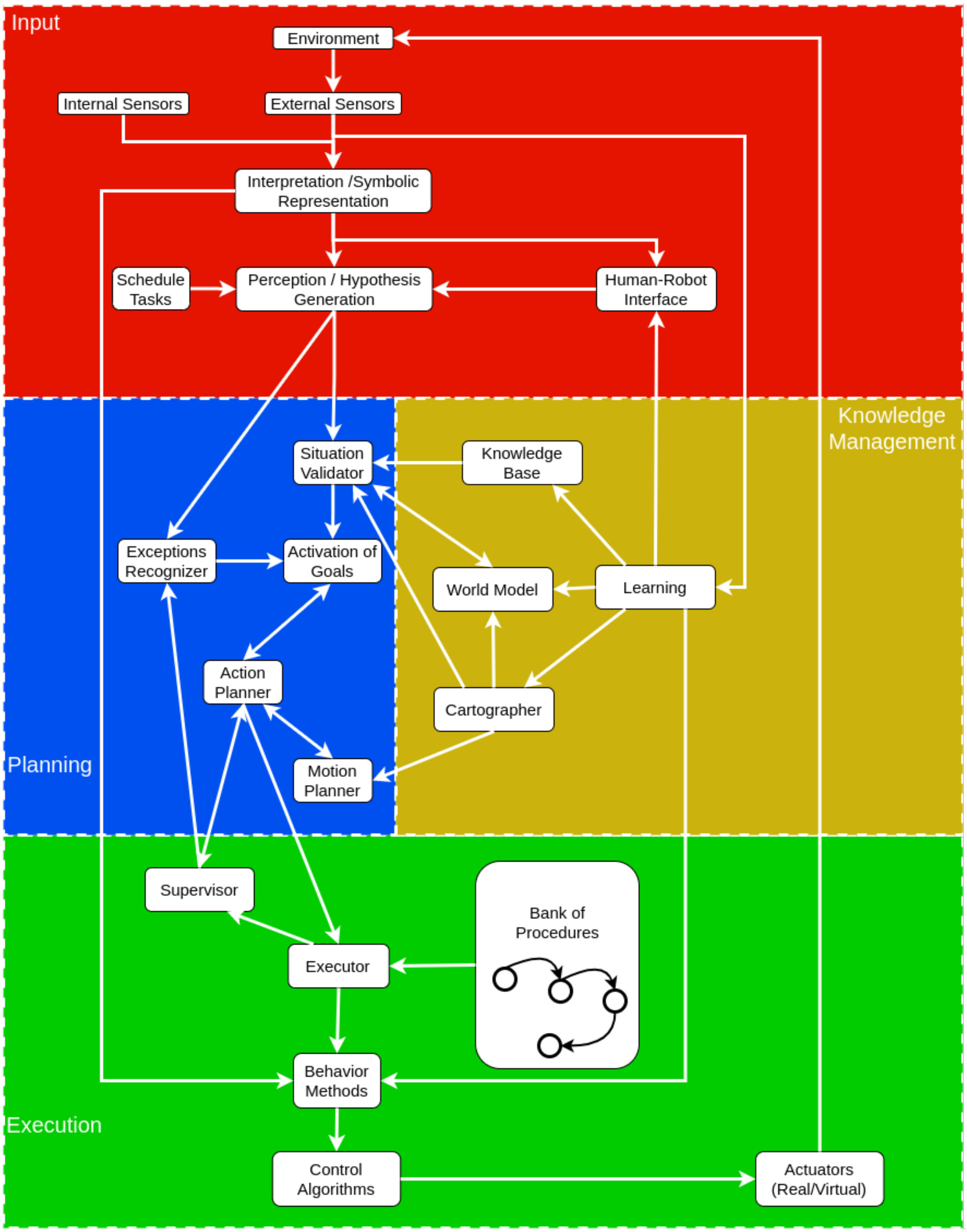

Fig. 2: VIRBOT structure

6.3 Navigation

We use a navigation system developed at the Biorobotics Laboratory. This system has been tested before in robot Justina in @Home league and has showed a better performance than ROS navigation stack. It uses A* algorithm for path planning, AMCL and odometry for localization, and a combination of Point Cloud-processing with the measurements of the LiDAR for obstacle avoidance. This system has been used in previous competitions for the robot Justina. It is programmed using mostly C++ and executed by the middleware ROS.

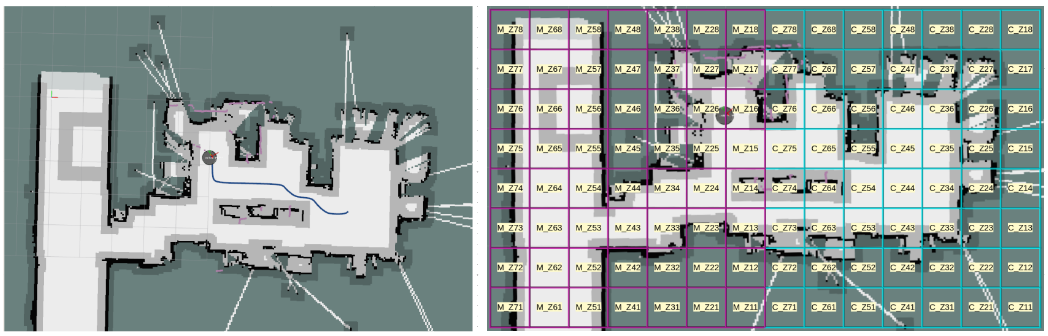

Fig. 3: Navigation Path

We have established the zones over the laboratory's map, to be able to navigate using a nomenclature similar to the one used in the competition.

6.4 Image Processing

We aim to use the image processing of the RGB-D camera to be able to align with the MPS and to the pieces to grasp them, so far we have been working with color segmentation using OpenCV packages and combining algorithms to obtain the centroid of the pieces, and we also use the Point Cloud to localize it in the workspace and defining its coordinates on the map, as well as exclude some of the noise generated in the segmentation. It is programmed in both C++ and Python and the communication is done through ROS topics.

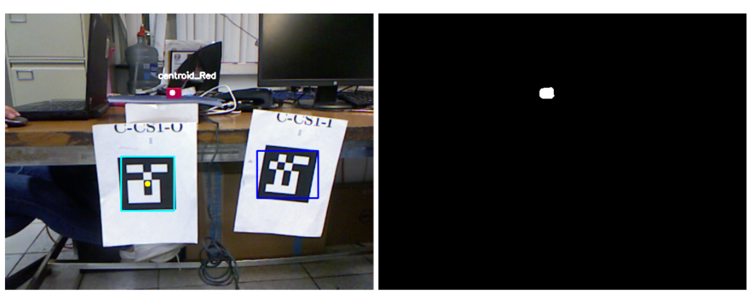

Fig. 4: MPS finding and Color segmentation

In our sequence, we first find the ARUCO tags and focus on the MPS of interest, then find the piece trough color segmentation (Figure 4), using the Point Cloud, we made the homogeneous transformation with respect to the robot and assign a static mark in the robot's map (Figure 6).

MPS alignment For the alignment to the stations, we are using a combination of filters and algorithms to be able to find planes and lines that show us the inclination of the station referring to the robot and minimizing that angle with base movements.

Fig. 5: Aligning with MPS

Gripper alignment Once the robot is as aligned with the stations the robot needs to align its gripper with the input of the conveyor belt, using Image-Processing and Point Cloud-Processing in an Active-Vision process.

Our intention is to analyze the visual information that the robot has received from the environment and change the position of the camera to be able to optimize the route that it should follow to grasp the piece, using a utility function to optimize the pose of the camera[3].

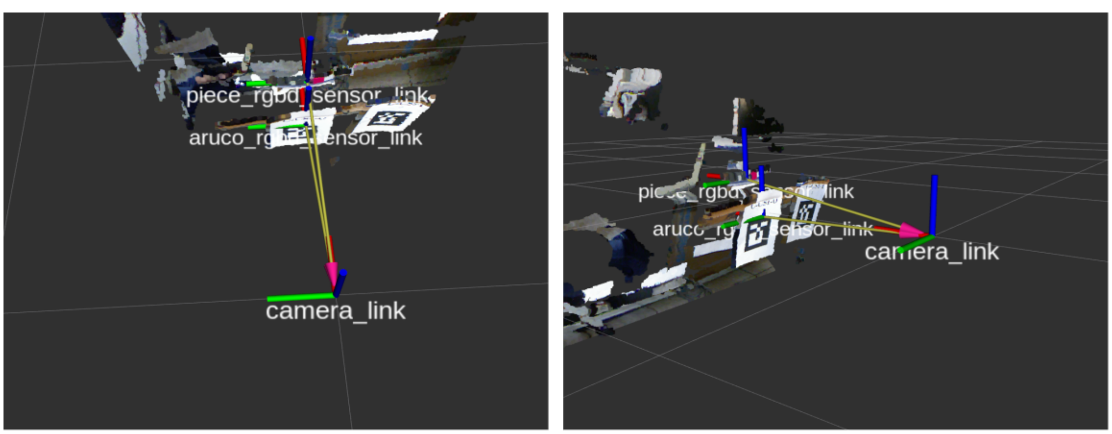

Fig. 6: Piece position

7 Conclusion

Being our first time in this League of the competition, we are anxiously expecting to tests our efforts in a competition-like environment, where we can see which of our approaches are suitable with the requirements of the League and which ones we must put more effort into. Also compare the performance of the previously developed software for the other categories of the RoboCup and how well it accommodates with the Logistics League. This first time will be the basis of the work to be done in the next years and the perspective that the Logistics division in the Biorobotics Laboratory will take.

References

[1] Peter Kohout et al. "A multi-robot architecture for the RoboCup Logistics League". en. In: *Elektrotech. Inftech.* 137.6 (Oct. 2020), pp. 291–296. ISSN: 1613-7620. DOI: 10.1007/s00502-020-00826-5. URL: https://doi.org/10.1007/s00502-020-00826-5 (visited on 10/17/2022).

[2] *Protocol Buffers*. en. URL: https://protobuf.dev/ (visited on 03/23/2023).

[3] Loris Roveda et al. "Robot End-Effector Mounted Camera Pose Optimization in Object Detection-Based Tasks". en. In: *J Intell Robot Syst* 104.1 (Jan. 2022), p. 16. ISSN: 0921-0296, 1573-0409. DOI: 10.1007/s10846-021-01558-0. URL: https://link.springer.com/10.1007/s10846-021-01558-0 (visited on 03/28/2023).

[4] Jesus Savage et al. "Semantic reasoning in service robots using expert systems". In: *Robotics and Autonomous Systems* 114 (2019), pp. 77–92. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2019.01.007. URL: https://www.sciencedirect.com/science/article/pii/S0921889018302501.

[5] Technical Committee 2012–2022. "RoboCup Logistics League Rules and Regulations 2022". English. In: (Apr. 2022), p. 48. URL: https://ll.robocup.org/the-rcll-competition/page-3-2/ (visited on 11/23/2022).