

Perception and Navigation in unknown environments: The DARPA Robotics Challenge

Eduardo J. Molinos, Ángel Llamazares, Noelia Hernández, Roberto Arroyo, Andrés Cela, José Javier Yebes, Manuel Ocaña, and Luis Miguel Bergasa

Department of Electronics, University of Alcalá, Madrid (Spain) {`emolinos`,
`allamazares`, `nhernandez`, `rarroyo`, `acela`, `jyebes`, `mocana`,
`bergasa`}@depeca.uah.es

Abstract. This paper presents different techniques to achieve the tasks proposed in the DARPA (Defense Advanced Research Projects Agency) VRC (Virtual Robotics Challenge), which entails the recognition of objects, the robot localization and the mapping of the simulated environments in the Challenge. Data acquisition relies on several sensors such as a stereo camera, a 2D laser, an IMU (Inertial Motion Unit) and stress sensors. Using the map and the position of the robot inside it, we propose a safe path planning to navigate through the environment using an Atlas humanoid robot.

Keywords: Humanoid Robots, Navigation, Mapping, Perception, DARPA Robotics Challenge

1 Introduction to Virtual Robotics Challenge

The DARPA Robotics Challenge [1] is an open competition focused on the development of humanoid robots with the aim of operating in rough terrain and austere conditions using human tools commonly available in real environments. The Virtual Robotics Challenge is a preliminary phase of the competition carried out in simulated scenarios and using open source tools like ROS (Robot Operating System) [2] and Gazebo [3]. Three different scenarios are proposed: get into a vehicle and drive it, walk through rough environments and manipulate unknown objects.

We have participated in this challenge as SARBOT Team [4], which is composed of three universities (Universidad Politécnica de Madrid, Universidad Carlos III and Universidad de Alcalá) and a research centre (Centro de Automática y Robótica). Universidad de Alcalá group is focused on the perception of the environment, mapping abilities and navigation planning.

2 VRC Proposal

The Virtual Robotics Challenge is divided into three scenarios with multiple tasks that have to be solved. We have divided the scenarios into simpler tasks so our multidisciplinary team could work on it in an efficient way.

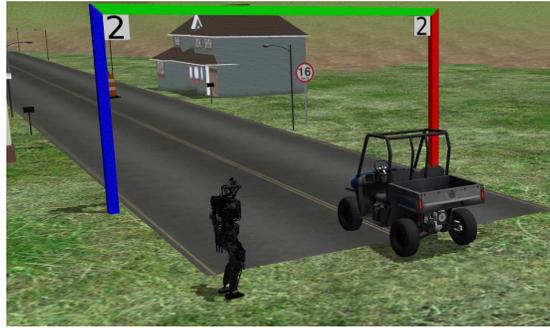


Fig. 1. Scenario 1. Atlas robot facing the car.

The Atlas Robot [5] is used to fulfil the tasks in all the three scenarios. This humanoid robot is equipped with several sensors, such as: stereo camera on the head of the robot, a Hokuyo UTM-30LX laser mounted into a rotating platform that can turn 360 degrees with an integrated IMU, an IMU in the pelvis of the robot, force sensors in the robot's foot plants, stress sensors in all the joints of the robot and a pair of Sandia hands (each of one equipped with a camera and force sensors).

Scenario 1 is mainly focused on the task of driving a car. In order to complete this task, we have to localize the robot into the starting pen, find the way out, detect the car and get close to its driver door, get into the car, start the car, drive it through the gates of the road, stop the car, get out the car and walk through the last gate. Figure 1 shows the Atlas Robot in the environment facing the vehicle.

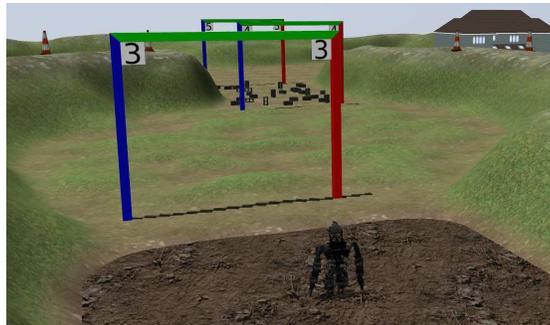


Fig. 2. Scenario 2. Atlas robot walking across the mud pit.

Scenario 2 is mainly focused on the task of walking in difficult terrain. As in the first scenario, the robot is initially into a starting pen and has to walk through the exit. Then, it has to detect the first gate and walk through it, go

down a ramp into the mud pit, walk across the mud pit in the direction of the second gate, go up the ramp to get out the mud pit, walk in a rough terrain, cross the third gate and finally walk across a terrain full of obstacles. Figure 2 shows the Atlas Robot walking across the mud pit.

Scenario 3 is mainly focused on the manipulation task. The robot must get out the starting pen, detect a table with a hose on it, approach to the table, grab the hose, connect it to a standpipe located on the wall and open a circular valve. Figure 3 shows the Atlas Robot facing the table to grasp the hose.

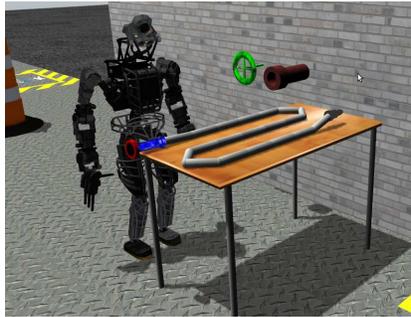


Fig. 3. Scenario 3. Atlas robot looking at the hose.

3 SARBOT Team Proposal: Perception and Navigation

Some tasks are common for all the proposed scenarios by the Virtual Robotics Challenge such as navigation and localization, although the goals of the navigation are different for each of them. The challenge sets some objects that can be recognised by analysing some of its characteristics: the gates, the starting pen, the vehicle, the table, and so on. We propose a combination of different methods based on camera and laser to identify the objects in the environment and plan the goals to go.

Hereafter, we explain different methods that we use to achieve the different tasks:

3.1 Mapping

We use the laser mounted on the head of the robot for the mapping task. We use a 3-D laser point cloud representation of the environment instead of the commonly used 2-D representation. This 3-D representation of the environment is obtained by rotating the laser 360 degrees and knowing its position from the data of the IMU mounted on the laser. Usually, the point cloud could be obtained employing the disparity map available if a stereo camera is considered,

but in VRC the accuracy of this point cloud is acceptable enough only when the object has rich textures. However, this is not the case for the VRC simulated environments. Then, we discard the disparity map in favor of the 3D data from the laser.

To ensure the consistency of the 3-D maps, we transform the point cloud obtained from each measurement into a discrete representation based on Octomap [6], which is a probabilistic volumetric representation of space based on Octrees that allows to represent a full 3-D environment minimizing memory requirements. This representation along with an effective movement model of the robot (obtained by the IMU data and stress sensors) give us the robustness needed to work without a SLAM (Simultaneous Localization And Mapping) system [7]. Consequently, the computational cost is also reduced with this approach. Figure 4 depicts a 3-D map of the task 2 environment and an image of the environment.

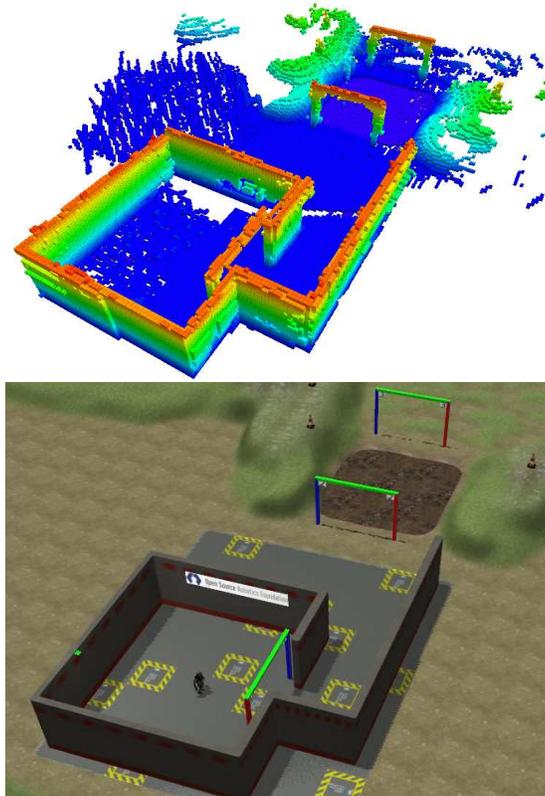


Fig. 4. 3-D Map of Task 2 Environment.

3.2 Navigation

Since navigation into a 3-D environment is a computationally expensive task and the Atlas Robot has a limited range of movement in the z axis, we can transform the built 3-D map into a 2-D representation, which allows to plan the optimal way to be followed by the robot saving computational cost. We use the heights difference from one 3-D block to another to mark a 2-D cell as occupied or free. The minimum height difference to be marked as occupied is calculated knowing the prior abilities of the robot to walk through different heights.

Once we have built the 2-D map from the 3-D map, we are able to calculate the optimal route to reach the goal. To do so, we use a navigation algorithm for humanoid robots [8] which uses an ARA* (Anytime A*) Planner [9] to plan the path to be followed by the robot. This algorithm searches the path from the robot to the goal avoiding rough turns at the starting of the movement and employs a cost function based on the Euclidean distance between initial pose and goal and the number of steps that the robot has to follow. We configure this algorithm with a discretisation of steps that the robot can follow allowing it to walk straight, turn and even walk backwards. Figure 5 shows the 2-D map representation where the steps to follow are marked in different colours: steps for left foot in red and steps for right foot in green. The goal is represented as a red arrow.

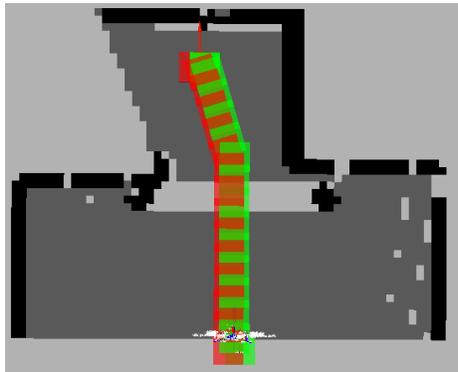


Fig. 5. Navigation. Step-by-step route

3.3 Goal Recognition

Once the navigation framework is set, recognition of different goals is needed to complete the tasks. There are different objects that can be recognised in the environment of VRC scenarios, following we show the techniques that we use to recognise them and plan the robot's navigation:

Gate recognition: an algorithm based on a mono-camera colour detection is employed to recognise the different gates in the environment, taking into account that the gate's upper post is green, the left post is blue and the right post is red. We train a Bayesian classifier [10] from a set of sampled images in a resolution of 800 x 800 pixels. The colour differentiation is made by employing an optimization based on Mahalanobis distance using HSV (Hue, Saturation, Value) channels, because this colour space allows a better discrimination between each type of post, attending to the hue and saturation characteristics. Then, we apply a skeleton function to ensure that all the posts are connected and the gate is fully recognised. Moreover, we use a SVM (Support Vector Machine) classifier implemented in LibLinear [11] and based on HOG (Histogram of Oriented Gradients) [12] features to identify the number of the gate and ensure we are detecting the proper gate. In this case, HOG is applied as a powerful shape descriptor which allows to differentiate the varied number silhouettes and classify them correctly. Figure 6 shows the gate recognition, its orientation with respect the robot and the number 2 detected on the gate.

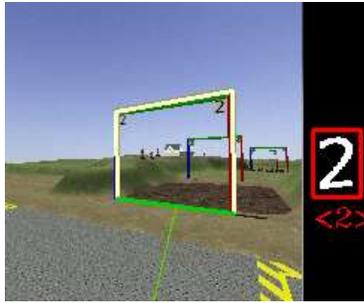


Fig. 6. Gate and number recognition.

Getting out of the starting pen: we have considered two different scenarios: depending on whether the first gate and the exit is on the line of sight of the camera or not. If the robot can see both sides of the gate (red and blue posts) in the image it means that the robot is looking at the exit, so the goal is the furthest point in the map in the robot's field of view. In case that only one or none of the posts of the gate are detected in the same image the robot turns until both posts are recognised. If both posts cannot be detected in the same image the robot is at the side of the exit and it moves toward the centre of the starting pen and turns until is facing the exit.

Once the robot is in the exiting corridor, we need to plan a path to exit it. Knowing that the corridor has only one exit and the entrance to the starting pen, we use an occupancy map where the upper posts of the gates and the walls are represented as occupied. This way, the entrance to the starting pen

is virtually blocked and the goal is calculated as the half distance from the furthest reachable point. Figure 7 shows the map built while exiting the starting pen where black cells represent obstacles detected above the robot, dark grey cells represent detections at other heights and light grey cells represent unknown cells. The outgoing path is depicted in green and red steps, the initial pose of the robot is represented as a pink rectangle and the pink arrow represents the final goal.

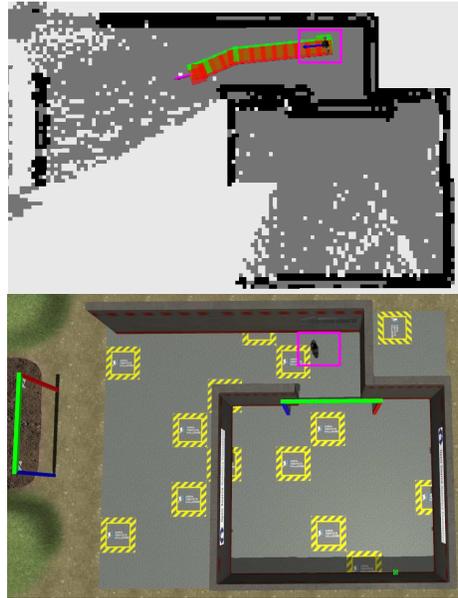


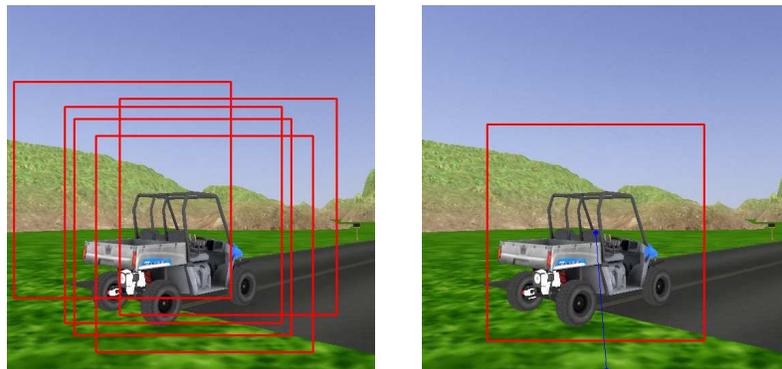
Fig. 7. Atlas robot getting out of the starting pen.

Approaching to a gate: the gate is detected using the previously explained gate recognition algorithm and the robot turns until it faces it. To be able to calculate the goal and plan the robot movement, the correspondence between the gate detected with the camera and the 2-D map representation is needed. To do so, occupancy map is built where occupied cells represent heights above the robot (to detect the gate upper posts). These cells are labeled looking for connected blobs in the robot's field of view corresponding to the gate. The goal is calculated as the blob's centroid and its direction as the normal vector of the gate.

Getting close to the table: assuming that the table is about one meter above the floor, we build a map where occupied cells represent objects between 0.8 and

1.2 meters high. The connected objects are classified rejecting the smallest ones. Then, the nearest object to the robot similar in size to the table is selected. The centroid of the table is used as a goal to get close to it. Once the robot is near the table the hose can be recognised by the cameras and its position can be obtained to grasp it.

Approaching to the car: an SVM classifier based on LibLinear [11] is employed with the aim of detecting the car and its orientation in the camera image. This classifier uses a combination of appearance descriptors based on HOG and GCH (Global Color Histogram) [13], with the aim of taking into account the shape and color of the car for the classification. At first, a grid is created for classifying the different parts of the image and a sliding window algorithm is applied to obtain the hypothetical places where the car could be located, as shown in Figure 8(a). After this, an average of the hypothetical positions of the car is carried out, obtaining the final position and its orientation in the image, as can be observed in the example presented in Figure 8(b). Once the car position is identified, the robot turns until it faces it. A map is created where the occupied cells represent objects above the floor and under 2 meters height (the height of the car) and the goal is set as the obstacle centroid to face the door and be able to get into the car.



(a) Hypothetical places detected after the sliding window classification. (b) Final car position detected (in red) and orientation estimation (in blue).

Fig. 8. Car detection and orientation estimation.

4 Conclusions

We have competed in the DARPA VRC by means of dividing complex tasks into simpler ones. In this work different techniques to navigate and map an unknown

environment using heterogeneous sensors to recognise objects are presented. We have also demonstrated that a full 3-D environment can be efficiently reduced to a 2-D map to plan an optimal path that a humanoid robot can follow.

Acknowledgements

This work has been possible thanks to SARBOT Team, formed by Centro de Automática y Robótica, Universidad Politécnica de Madrid, Universidad Carlos III and Universidad de Alcalá. This work has been funded by the “Ministerio de Economía y Competitividad” through TIN2011-29824-C02-01 and TIN2011-29824-C02-02 (ABSYNTHÉ project) as well as by the Comunidad de Madrid through the project Robocity2030 (CAM-5-0505/DPI000176).

References

1. “The Robotics Challenge: <http://www.theroboticschallenge.org/>.”
2. “Robot Operating System (ROS): <http://www.ros.org/>.”
3. “Gazebo Simulator: <http://www.gazebosim.org/>.”
4. “SARBOT team: <http://www.sarbot-team.es/>.”
5. “Atlas Robot: http://www.bostondynamics.com/robot_atlas.html.”
6. A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees,” *Autonomous Robots*, 2013.
7. H. Durrant-Whyte and T. Bailey, “Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms,” *IEEE Robotics and Automation Magazine*, vol. 2, p. 2006, 2006.
8. A. Hornung, A. Dornbush, M. Likhachev, and M. Bennewitz, “Anytime Search-Based Footstep Planning with Suboptimality Bounds,” in *Proc. of the IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2012.
9. M. Likhachev, G. Gordon, and S. Thrun, “ARA*: Anytime A* with Provable Bounds on Sub-Optimality,” in *In Advances In Neural Information Processing Systems 16: Proceedings of the 2003 conference (NIPS-03)*, MIT Press, 2004.
10. H. Zhang, “The optimality of Naive Bayes,” in *FLAIRS Conference*, pp. 562–567, 2004.
11. R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “LIBLINEAR: A Library for Large Linear Classification,” *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.
12. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 886–893, 2005.
13. C. L. Novak and S. A. Shafer, “Anatomy of a color histogram,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 599–605, 1992.