DESCRIPTION

INTRODUCTION

The JdeRobot-Academy framework for learning robotics is presented on this talk. It is open source, based on ROS middleware and on Gazebo, and uses Python as programming language for the robot logic. It is intended to improve the educational tools available to learn robotics, and aims to bring the field of robotics to the university engineering students in a practical and appealing way, emphasizing in the programming of the robot intelligence more than on its construction.

DESIGN

The framework consists of a set of independent exercises, related to successful robotic applications in society (autonomous cars, vacuum cleaners, drones, logistics...). Each exercise is seen as the combination of three layers: physical robot, middleware and application. The lower layer is the physical robot which is aimed to perform some task in a certain environment. The JdeRobot-Academy exercises do not focus on a single specific platform, but use different ones (TurtleBot, cameras, Formula1 car, Roomba, etc.), both simulated in Gazebo or real robots when available. The intermediate layer are the corresponding drivers that give software access to the sensors and actuators of the robot. ROS has been selected for that. In the application layer lies the student's code and algorithms.

Robotic systems are complex. In order to make them easier to the newcomer students an academic ROS node has been created for each JdeRobot-Academy exercise. It

a) provides a programming interface for accessing the sensors and actuators of the robot with simple Python methods (HAL API);
b) provides an specific Graphical User Interface for that exercise, which is useful for debuggin purposes, and a simple programming interface for it (GUI API);
c) provides a single file template to be filled by the students with their code; (d) includes code that solves auxiliary tasks, for instance a temporary skeleton based on a continuous loop of iterations for reactive behaviors.

These academic nodes host the student's code which analyzes the sensory data and makes decisions of action, planning if necessary. It can be connected to the real or simulated robot interchangeably just using different configuration files, but without changes on the code itself.

The use of standard libraries and tools in robotics such as OpenCV or PCL is also encouraged when programming the student solution for each exercise.

EXERCISES

The current exercise collection is available at Github[1] and the practices can be grouped in four sets[[2]]: autonomous cars (like [[3]] and [[4]]), computer vision, drones and mobile robotics.

Two recent and illustrative exercises will be also detailed in the talk. First, the FollowLine exercise, where the goal is that a Formula-1 car follow the red line in a circuit in the shortest time [[5]]. The student's solution will

[1] https://github.com/JdeRobot/Academy
[2] https://jderobot.org/JdeRobot-Academy
[3] https://www.youtube.com/watch?v=uaufrpVLdRA&t=182s
[4] https://youtu.be/q6G6BHqljP4
[5] https://www.youtube.com/watch?v=QGO9oaoBVoA

extract information from the pixels of the on board camera of the robot and will order the corresponding motor commands for the driving wheel and to the accelerator pedal. The student has to learn color filters and ad-hoc processes that extract some information, as whether the car is over the red line or not or somekind of deviation. A typical solution includes also a case based reactive controller and one or two PID controls. Its graphical interface facilitates the visualization of the processing performed on the image.

Second, in the vacuum cleaner with SLAM exercise the goal is to implement a navigation algorithm to cover the largest area of a known house [[6]]. The map of the house is provided in a file and the robot is assumed to be precisely located all the time. The solution of the exercise begins with the planning of the cleaning route, creating a grid on the map to carry out a zigzag path by checking on each pixel if the surrounding cells belong to any obstacle, if they have already been visited, or if they are still to be cleaned. The graphical interface shows the map of the house and the path followed by the robot. An an automatic evaluator has been developed for this exercise, it connects with the simulator and provides a score dependending of the size of the cleaned area after a given time.

INSTALLATION

The installation of the framework has been prepared carefully and made very simple on purpose. It only requires the installation of the standard ROS Kinetic packages, the official Gazebo-7 packages, some configuration files and cloning the JdeRobot-Academy github repository.

BALANCE

The JdeRobot-Academy framework has been successfully used on three degree subjects and one postgraduate course at Universidad Rey Juan Carlos. It is open to further collaborations, extensions or modifications. It will be also used in the PROGRAM-A-ROBOT[7] Competition inside IROS-2018[8]. Some demonstration videos of the available practices can be seen in YouTube[9].

FUTURE LINES

First, the JdeRobot-Academy team is working on adapting seven existing drone exercises to ROS middleware. They currently run using ICE middleware and Parrot drivers but we are migrating to MAVros as the drone middleware for them. Second, an update is planned towards ROS-Melodic and Gazebo 9 in the near future. Third we are also working on creating a web release of the framework allowing the students to use a WebIDE and program the robots from the browser using Jupyter.

[6] https://www.youtube.com/watch?v=sUT5ru4Ew_E

[7] https://jderobot.org/Program-A-Robot-2018

[8] https://www.iros2018.org

[9] https://www.youtube.com/channel/UCgmUgpircYAv_QhLQziHJOQ