br_swarm Documentation

Release 2.0.2

Pototo

October 11, 2013

CONTENTS

This is the main documentation for the Brookstone Rover project. The main porpuse of this project is to create an autonomous system capable of recognizing hand gesture commands from a human user. These hand gesture commands include point goal (e.g. pointing to an object that the human user need, such as a tool), and intention recognition (grabing, reaching towards a tool). Once the system recognizes the gestures or intensions, the robots can then move to the desired goal point, or move towards the tool the human needs, and bring the tool to the human.

Requirements:

- Linux operating system (so far tested only with Ubuntu 12.10.02)
- Robot Operating System (ROS - Hydro or above to connect to multiple robots)
- Python 3.0 or above (necessary for connection to multiple robots with ROS)
- Brookstone Rovers (so far only tested with v1.0)
- Kivy GUI library
- A wifi dongle (Network interface Card or NIC) per robot

Contents:

CHAPTER

ONE

PROJECT SUMMARY

1.1 Goals Achieved

Goal1: Write Python code to connect to a single robot (including) acquiring image data and movement control.

Goal2: Separate code in nodes, and run them individually by using ROS.

Goal3: Publish robot data in the network (such as image data) so that other nodes can utilize that data (for optical flow, etc.).

Goal4: Created a simple Kivy GUI in order to display published image from the rover and drive the robot around.

1.2 Future Goals

1. We want to change some publish/subscriber implementation to Service/Client implementation. This might be more efficient than to permanently publish data over the network, especially when the data is not needed by any node.
2. Implement PWM code on the br_control node.
3. Implement a Optical Flow node to be used for path planning (including finding out if running out of battery based on the optical flow). This node shall be written in C++ for performance. ROS/OpenCV already have Optical Flow code.
4. Connect to multiple robots autonomously. This could be achieved reading the out of each NIC after they been connected to the robot Python can read this out.
5. Implement hand and gesture recognition code using ROS and the Kinect. There is already code in ROS and OpenNI to achieve this.
6. Implement point to goal features.
7. Send robots to the pointed to location.
8. Select an specific robot and send it to a specific goal, while the other robot (s) stay in the same place or are sent to another specific location.

CHAPTER

TWO

DOCUMENTATION FOR THE CODE

CHAPTER

THREE

BR_CONTROL CLASS

This file is used when you want to control a single robot i.e., the first rover that computer connects to

CHAPTER

FOUR

BR_CAM CLASS

This node deals with the rover's camera. It gets image data and publishes is so that other nodes can use it.

CHAPTER

FIVE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

5.1 A Subpoint

This is my idea.

5.1.1 A subsubpoint

This is a smaller idea.

PYTHON MODULE INDEX

b

INDEX

B