Zijun Xu, Roy De Jesus, Nic Williams

Professor Hayes

CSCI-3302

University of Colorado Boulder

19 December 2017

Teaching Robots How to Love

Abstract

During the semester we've been working on the project *Teaching Robots How to Love*. As the name implied, our work is based on a robot called Sparki and teaching it to see color. There are three parts of our project. 1) Language recognition. 2) Color/shape filtering. 3) Robot kinematics. While only part 3 is done in the Sparki, part 1 and part 2 are on the computer side which brought unique challenges. We use an external camera as Sparki's sensor and our goal is to connect the camera, computer terminal, and Sparki together. The final result should be: In the terminal, the user types "Can you go get the red ball?", then our language code gets the keywords of the sentence, in this case it should be "red". Then "red" will be passed to color filtering code, which will make use of the camera and find the location of the red object. It determines the grid id where the object is, passes it to Sparki's kinematics code through the serial port. Finally, the Sparki uses IK code to reach the goal grid, and capture the object. The result turned out to be a success, but errors could happen when the light is dim, or when the object is not sitting at the center of the grid. Since the Sparki's IK code has a final angle pose, it is possible that the Sparki knocks the object away hence fails to capture the object.

Background

Open Computer Vision, or openCV, is a computer image detection program that uses pixel detection to analyze and modify. Many experiments have been completed using this program; especially in the robotics field. Videos in youtube and project display a short, but deep, history of openCV.

openCV has been implemented by many YouTubers. This project was built mainly off the examples given by a YouTube channel, sentdex. He gives great examples and a good background of the importance.

Methods

Our process began with having Sparki understand the language we wanted. In order to do that we had to parse out the language that we type into the terminal. This utilizes the NLTK python library, which is a powerful language parsing tool. Our implementation takes the raw input from terminal and type-casts it as a string. Then the library splits the string into “tokens” (words that can be searched in different ways by type i.e. nouns, verbs, adjectives, etc.). It’s stored in an array, and then it’s cleaned up a bit. First we removed any “stopwords” (pronouns, conjunctions, punctuation, anything not necessary for object/color recognition), then we sterilized the words (ensured they were all lower case, properly spelled, etc.. Once the language had been reduced to key tokens only, we do a quick array search for any term matching our query (objects and color), and return a color code to parse to the camera so it knows what to seek.

Once our language code pass a color name to our color code, it will make use of the external camera and find the corresponding object. The camera vision is divided into 16 grids (which match our actual map). It will decide which grid is the object at, and pass that grid number to sparki through serial port.

In order to send information from terminal to Sparki, we used python library Pyserial. Pyserial allow us to send information through serial port from terminal. It is useful when connecting Sparki to the terminal. We can just simply connect the Sparki and laptop through a cable (or use the bluetooth module) and then we are able to send information(grid id) to Sparki through terminal.

Results

The project result is successful. We were able to get things working together (camera, terminal, Sparki). However we also had a few problems. 1) If light is dim, it could be hard for camera to recognize/distinguish colors. 2) Sparki couldn't read series of numbers (it will only read the first bit of the number, for example, if color code pass the value 14, Sparki can only read 1), but it were able to read series of letters, like "fourteen". 3) The IK code isn't perfect. Since it has a final angle pose, Sparki sometime knocks the object away and fail to capture it. 4) Language parsing and word reconstruction becomes very hard when you add numbers, dates, or punctuation between keywords. So if you typed "gr-een" the parse would fail. However, this was a lesson in how to build more secure computer passwords (as token reconstruction is an NP-complete problem!!)

Here's a quick video on it: https://youtu.be/lbfPLBYJUpA