Just Exclusively for Fetch: A Study in Mobile Object Retrieval

Josh Peterson  
 Computer Science  
 Utah State University  
 Logan, Utah, USA  
a02279843@usu.edu

Dylan Doelling  
 Computer Science  
 Utah State University  
 Logan, Utah, USA  
a02371420@usu.edu

Anthony Lasley  
 Computer Science  
 Utah State University  
 Logan, Utah, USA  
 a01644627@usu.edu

Mitch Parsons  
 Computer Science  
 Utah State University  
 Logan, Utah, USA  
a02088785@usu.edu

ABSTRACT

Automatic object retrieval is a very promising field of study and has an array of useful applications. Mobile retrieval systems are used for a variety of applications such as delivering medication in hospitals and retrieving objects in hazardous conditions. This project attempts to develop a mobile retrieval system by programming a robot to retrieve a ball using OpenCV, blob following, and inverse kinematics. In a simulation we developed a robot that successfully picks up a cylinder.

1 Introduction

We are using the Jetson Nano SLAM Lidar Car from XIAOR Geek, which we nicknamed JEFF (Just Exclusively for Fetch). JEFF is a skid steer robot, with a claw, camera and lidar sensor. JEFF is a good entry level robot for object retrieval. Our work can be expanded to work in more unique circumstances such as retrieving medication in hospitals or helping disabled persons retrieve household items, or simply playing fetch. JEFF will use OpenCV to run image detection to find a ball within JEFF’s field of vision. JEFF will then use a path finding algorithm to approach the ball with the onboard camera and lidar sensors. JEFF will then use inverse kinematics via the MoveIt package to pick up the ball. For this project we will be picking up a can; because, picking up a ball adds more complexity. Unfortunately, we could not get JEFF to work properly, so we decided to use the Webots robot simulator to accomplish the task

2 Methodology

2.1 Building JEFF

We started off the project by building JEFF. This was an optional requirement. We built the robot because we wanted more experience with the hardware side of robotics. Building JEFF went generally smoothly. The instructions were not very clear; because, the instructions were translated from Chinese. There were also screws were improperly manufactured, which led to difficulties in the building process. In the end, we were able to finish building JEFF, and got the robot working with a remote control.

2.2 Image Recognition On JEFF

In order to know where JEFF needs to go, we needed to implement the ball detection. We found a tutorial detailing how to do image recognition for a ball using OpenCV [2]. We first implemented the image detection on a laptop to make sure the code worked using a webcam. In order to get the program working on JEFF we had to set the foundation for running the software on JEFF. First, we attempted to get Firefox downloaded onto JEFF, so we would be able to download the source code. Downloading Firefox took surprisingly more effort than expected. We were not able to get Firefox downloaded while connected to the school’s Wi-Fi. This required us to take JEFF off campus to download Firefox there. Then, we had to get apt-get to work before we could download Firefox. Once we had Firefox downloaded on JEFF, we needed to get a text editor downloaded onto JEFF, so we could program on JEFF more easily. Downloading the text editor again took significantly more effort than expected; because, working on JEFF was our first time working with an ARM64 architecture. We ended up having to run some terminal commands, so we could connect to a windows server and download the correct version of the text editor. We then ran into the problem of trying to get the correct python packages downloaded onto JEFF. Initially JEFF didn’t recognize the pip command, so we had to get pip installed and working using apt-get. Using pip, we were able to install the correct python packages. After the setup process was finished, we were able to get ball detection program running on JEFF.

2.3 ROS On JEFF

In order to control our robot, we needed to use ROS in conjunction with our image recognition, but no one in our group had experience using ROS. We quickly discovered that ROS was more complicated than we anticipated. We spent a significant amount of time working on the robot with ROS, following tutorials, to try to control the movement of the robot. We were able to run a simple Hello World program utilizing ROS but had no success running the packages required for movement.

2.3 Things We Tried That Did Not Work

Our first attempt at setting up a python environment was trying to use anaconda. After first downloading and trying to run the wrong version of anaconda, we were able to install anaconda on Jeff. However, when trying to activate the base environment, the process would error out. The error was related to ARM64 not being compatible with a specific sub-package that was trying to run during the initialization process. After spending hours trying to figure out a way to solve this error, we decided the best solution was to abandon anaconda and go with a different package manager for python.

After we got image detection working, we moved on to getting JEFF to respond to commands. The provided instruction manual pointed toward the MoveIt package. That package is used in conjunction with ROS, which we were unable to get working. The instruction manual had us go through a tutorial to make sure the MoveIt package be recognized by ROS. The first part of the tutorial, which we got working, was the Hello World in ROS program. When we tried to call the MoveIt package there were many errors. One of the errors that we could not fix was figuring out why the package Catkin was not being recognized. The error gave us file paths of where the error thought Catkin should be installed and Catkin was in that specified area; however, MoveIt still did not work. We never figured out why Catkin was not recognized. We spent hours trying to fix this problem. We followed stack overflow posts and other online solutions; however, those solutions did not work.

We also went to Dr. Harper’s office hours and Dr. Harper suggested that we use pyserial to move JEFF. We watched some tutorials, but we never got pyserial working; because, pyserial never recognized the servos connected to JEFF.

2.4 Webots

With little remaining time, we decided to build our project in a simulated world rather than implementing the project using the real-world robot. We opted for the Webots simulation, and worked using the prebuilt Pal-robotics titanium robot, which has access to lidar and a robotic arm with a humanlike hand on the end.

Because we were learning an entire new framework, we had to relearn the system before we could make progress on the actual project. After spending time going through tutorials until we were comfortable with Webots, we began working on the project. We split the problem into two smaller problems and worked on each problem separately before combining the solution.

The first problem was using the lidar to approach the object which we wanted to pick up. The built-in lidar gave us the information needed to see in front of the robot. If the closest Item was on the left, we had the robot turn left, and if the closest item was on the right, we had the robot turn right. If the closest item was in the middle, we had the robot go forward and stop once the item was an arm’s reach away.

The second problem was gripping the pole. This was made more complex due to having a human-like hand rather than a two-finger pincher. To grip the pole all five fingers had to work in tandem. The most difficult part was the thumb. We opted to start be setting the arm and thumb before moving the robot, and then approach the pole. Once the pole was inside the gripping area of the hand, we instructed the robot to grip with the remaining four fingers and finish the gripping motion with the thumb. We then had the robot pick up the pole and continue moving forward.

3 Results

Using Webots and the prebuilt Pal-robotics titanium robot we were able to successfully program a robot to pick up a pole. This is close to our minimal viable product we talked to Dr. Harper about with using the actual robot, JEFF. The Minimal Viable Product was using JEFF to go to the ball and picking up a can. In Webots we used lidar to go to the ball. After that we were able to pick up a pole, and we got ball detection using OpenCV to work on JEFF. The code and videos are on Josh Person’s Github [1].

4 Conclusion

We learned a lot while working on this project. We gained valuable experience in some of the difficulties related to programming robotic projects, including hardware software interaction, interactions between sensors and motors, and learning new software technologies.

The hardware software interaction was a difficult hurdle which we were unable to overcome. There were multiple solutions which we tried to overcome this hurdle, but each solution had its own challenges.

The interaction between sensors and the different motors of the robot was a difficulty we experienced primarily in the Webots simulator. It was difficult to parse the information gathered from the lidar and use that information to determine whether to turn left, turn right, or go forward as well as how to move the various motors related to robot arm and hand.

Learning new software technologies takes a lot of time. Because of our groups relative lack of experience working with robots, ROS, and other packages, we experienced a lot of bottle necks and delays throughout the course of this project.

REFERENCES

[1] J. Peterson, “Joshualp2000/cs5510finalproject,” GitHub. [Online]. Available: https://github.com/joshualp2000/CS5510FinalProject. [Accessed: 15-Dec-2021].

[2] A. Rosebrock, “Ball tracking with opencv,” PyImageSearch, 17-Apr-2021. [Online]. Available: https://www.pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/. [Accessed: 15-Dec-2021].

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00