Homework 4 – Arrow Detection and Orientation Tracking

Submitted in partial fulfillment of the requirements for the course of

ENPM809T – Autonomous Robotics

By

Adheesh Chatterjee

Course Faculty
Prof. Steven Mitchell

UNIVERSITY OF MARYLAND
18 56

A. JAMES CLARK
SCHOOL OF ENGINEERING

INTRODUCTION

The aim of this project is to use OpenCV on the Raspberry Pi to facilitate object identification and subsequent robotic control/command, including writing data to file and analysis in Matplotlib. We first test the functionality of our ultrasonic sensor by recording distance measurements. Then, we build upon our object tracking algorithms to identify the presence and orientation of a green arrow Finally, the analysis of hardware performance limitations is performed.

APPROACH

To test our ultrasonic sensor, we first place an object 0.5m away from the pi. We then record a still image of the scene and using the ultrasonic sensor, record 10 successive distance measurements between the pi and the object. We finally calculate the average of the distances and print them onto the image.

For arrow detection, we first mask the image by converting it to the HSV scale to detect green colour. We blur the image using Gaussian blur. Then a contour of the detected region is found and the corners are identified. Next, we detect the orientation of the arrow based on the size of the histograms of the detected arrow, if the the max value in the histogram is greater than the width of the detected arrow, then it is pointing right and vice versa. A similar logic is used to check if the arrow is pointing up or down based on the columns of the histogram. This makes the detection algorithm more robust and less dependent on the corners. Finally, the arrow is moved around the screen and rotated. Lastly, the time taken per frame is tracked by writing to a .txt file, and the data is plotted as an x/y graph of processing time vs frame and as a histogram of processing time vs number of frames.

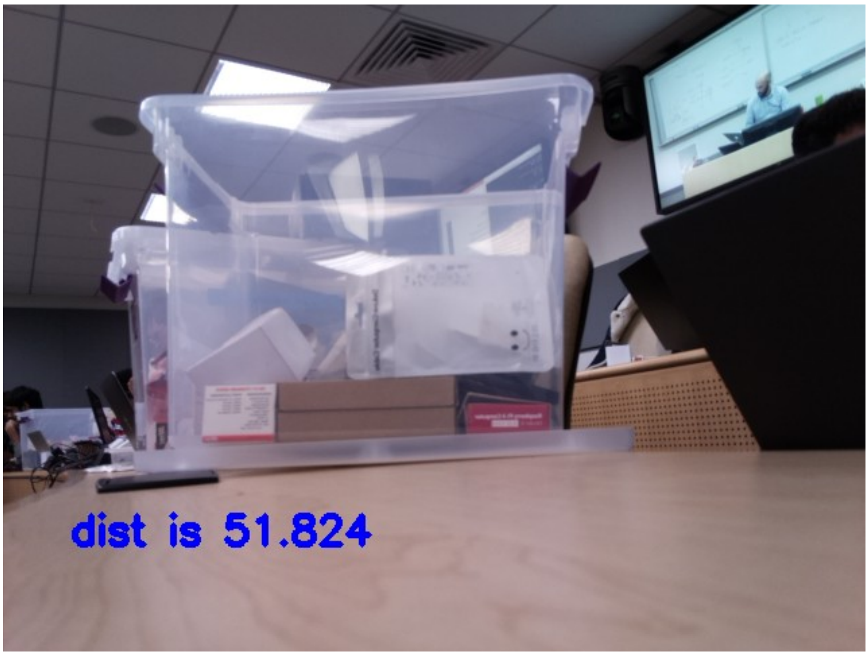
CODE AND OUTPUT VIDEO

The entire repository containing the problem statement and the data can be viewed on https://github.com/adheeshc/raspi-Arrow-Orientation

The YouTube output video can also be viewed on https://www.youtube.com/watch?v=Qow_cB72z0U&feature=youtu.be

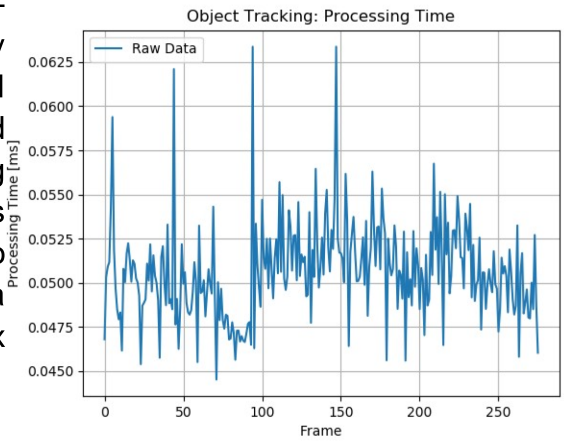
RESULTS

Ultrasonic Sensor Results



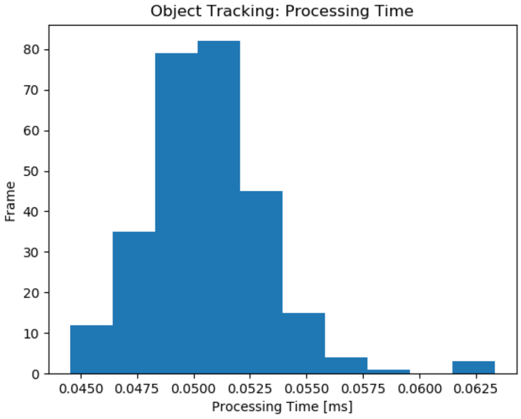
The resulting image for the sonar sensor readings is shown with the distance approximately 51.824m

Arrow Orientation Results



The processing times tend to be between 45 – 60 milliseconds. The image processing is pretty simple with very less computation required and so we get a good frame rate with good processing speeds. However, on increasing our computation as per our requirements this delay can cause problems when trying to control the robot. So multi threading might be a good idea especially when more complex object recognition is performed.

In our histogram plot, we see most of our frame rates are concentrated around the 0.05s mark which is a pretty good result which can we can try to improve as we add more computation


Object Tracking: Processing Time
Frame
Processing Time [ms]