Robot Vehicle System Research

This document reflects an effort to research alternative methods by which we will be able to implement our project requirements. Specifically, this document will research the possibility of using a Raspberry pi with a pi camera, and the OpenCV library to control our robotic vehicle system.

We would most likely be using the newest Raspberry Pi 3 Model B+, which would come with built-in wireless LAN, and BLE (<https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/)>. The board runs for about $30. There would not be a SD Card or a power unit, so for each Pi they would have to be purchased separately (SD card price depends on size & $10.99 respectively). There would also need to be a camera system that can connect to the raspberry pi (<https://www.raspberrypi.org/products/camera-module-v2/)>. It can either be the pi camera which is $24.99 or a cheaper webcam. It would ideally be a low power camera with enough resolution so process the images (i.e. it could be a usb webcam).

These components would be used to capture the image, and then once we have the image we can process it on the raspberry pi directly or on an application that controls our whole system. The library that I have familiarity with is OpenCV (<https://opencv.org/)>. The library is written in C++ and provides many functions and tools used for computer vision. There are also Python and Java wrappers that allow you to write code in those languages. There is also a lot more support and documentation for OpenCV as it has become more popular since computer vision is an increasingly popular topic. The current version of OpenCV is 3.4.3 and was released 8/29/18 so it is very current. The library would provide us with almost all the resources we need to perform collision detection and other features, such as calculating the position relative to the lead vehicle.

There are GPIO pins on the Raspberry Pi that we could use to control the robot vehicle, but ideally we will already have the vehicle built and will just need to figure out a way to interface with it. We would need to be able to send information to the unit as well as receive it.

Another alternative option would be to use the computer vision and image processing capabilities that can be included with the MATLAB program (<https://www.mathworks.com/products/matlab.html)>. This provides the users with a lot of built-in features that you would otherwise have to explicitly program in OpenCV. The student cost of the program would be $100 and then there is the $10 add-on for the computer vision module. If we are able to find the correct robotic vehicle, this may be the best system to use because it allows you to perform a lot of high-level actions and not waste time coding menial tasks.

The system using MATLAB would be contingent on the robotic vehicle, however for processing the data it is a good tool.