Samuel Choi
EcoCAR 3 Fall 2016
Role: ADAS/Innovation Team Member
December 1, 2016

Final Report

This semester was my first semester as a part of EcoCAR 3. After having only one person working on the computing related tasks for the team spring 2016, a wholly new team for ADAS and innovation was formed. Adjusting and getting the flow of work took a few weeks. After one of our team leaders went to a Mathworks conference in Massachusetts, our team had a solid understanding of what needed to be done. I worked mostly on innovation and occasionally helped with ADAS tasks.

For ADAS, our end goal is to have the car's computer vision system to be able to detect various objects from pedestrians, cars, lanes, different traffic signs, traffic lights, and any other object that will affect the behavior of the driver. The intent is to use this to have the car be in the optimal mode for energy efficiency. For example, if the car was approaching a stop sign, the car should ideally go to electric mode and prepare to recover energy during braking. During the semester, we have gotten detection of a few different objects to work using sample video files, and a couple of them have been merged together. In addition, we have the hardware, which consists of a mount with two webcams, ready to be tested for calibration in a car. The next step would be to merge all of the different object detections into one program, optimize the code in C if necessary, and find out if the given computer board is able to handle the program. A lot of progress has been made, but there is still a lot to do next semester.

In the innovation category, our team is proposing a program that will use map data on a set GPS route to optimize battery charge. The program would get various data including, coordinates, elevation, distance, average speed, average time, and change in angle for every segment traveled. Using this information, the car would know how to use its energy better. For example, if the hill is sloped downward, the car will know that it should attempt to recover the unnecessary kinetic energy. The unit that we will be using to handle all of this is a GPS device called TomTom. However, due to it coming so late, we had to create our first iteration of our algorithms elsewhere. I was responsible for getting the map and route data, and I wrote the code in MATLAB and I used data requested from Google maps. The next step is to convert all of our work into code to be used by the TomTom.

In the MATLAB code I have written, coordinates, elevation, change in pitch and yaw, and distances are retrieved. To run the code, simply open the main file, locationdata2.m and put the starting and end point as the inputs.

```
[out,out2] = locationData2(start, dest)
```

Here is how the code works in order.

- The code first gets a polyline from Google.
- Then the polyline is run in another function called googlePolylineDecoder.m, which converts the polyline into an array of coordinates. (extra MATLAB toolboxes required) This code is from the Mathworks website.
- Then elevation data is retrieved from Google for all of the coordinate points.
- The elevation data is requested at a rate of 100 coordinates at a time, because if all are requested at once, Google's servers get overloaded. Also, they are not requested individually as Google only allows 1000 free queries per day.
- The coordinates and corresponding coordinates are then run in another function called earthDistance.m, which them calculates the distance between each point as well as the pitch and yaw angles. Earth is not completely sphereical, so they use ellipsoidal formulae, distance formula, and trigonometry to calculate the data. The standard it's following is WGS (84), which is commonly used by GPS navigation systems. The tested distances are the same as what Google outputs.

Thanks for a great semester!