Christopher Estock

Object Oriented Programming

Homework 3

Intelligent Ground Vehicle – Preliminary Programming

This object and line detection program is designed to be implemented onboard an intelligent ground vehicle for the purpose of avoiding obstacles and remaining within boundaries outlined by white lines. The program (IGVC\_main.cpp) is written in C++ for the reason that its primary computer vision and image processing library, OpenCV, is also written in native C++. There are also Python and C implementations of OpenCV, but the C++ implementation will work well with an objective programming approach. Within the program, OpenCV is used to provide various image processing algorithms such as thresholding, masking, blurring, edge detection, and several drawing functions. For the vehicle to perform at an acceptable speed, these functions need to execute quick enough so that the obstacle avoidance and line detection systems do not introduce lag to the overall function of the vehicle’s main program.

There a several problems that need to be solved by the obstacle and line detection program. The primary problem to be solved is the white line detection. The obstacle detection acts as a failsafe for another system (LiDAR – Laser Radar – Light Detection and Ranging), so should be classified as a secondary feature. Using a set of three webcams (Logitech c920), this program must have the ability to discern white lines from light green grass and other colors. In order to perform this, a series of algorithms will be used to filter, threshold, mask, blur and remove noise from the webcams video feed. Most of these image processing algorithms are included in the OpenCV library, however, in order to correctly draw contours, rectangles and lines I must correctly implement several methods for determining the location, density and general shape of vectors of points – sometimes numbered in the thousands.

For now, the program is not objectified. At a point in the future when the image processing techniques are refined and confirmed to be operating successfully, the process of objectifying will begin. As a reference, the program will be required to work fluidly with the LiDAR component of the program being written by a classmate. When the computer vision (line, obstacle detection) system is finally interfaced with the LiDAR system, there will only exist a vector a points in which the vehicle cannot travel which encompasses all lines and obstacles. This means my program must simplify any detected lines or obstacles into a vector of points whose locations are relative to the current position of the vehicle.

Currently, the program’s dependencies are limited to the string, math, and standard library along with the OpenCV computer vision library. Another dependence is the custom header file ‘IGVC\_imgproc.h’, which is used to store the custom image processing algorithms that were made specifically for the intelligent ground vehicle. These algorithms are all custom derivations of various OpenCV algorithms combined with various math, string and vector functions. Aside from the libraries and header files, a video from past years’ winners is being used to test the image processing algorithms and should be included in the github repository named ‘IGVC\_test.mov’.

In order to build and run the program, the OpenCV (2.4) should be installed on the target machine. This program was written in Visual Studio 2012 and requires ‘IGVC\_imgproc.h’ to be in the same project as ‘IGVC\_main.cpp’. In the github repository, a text file listing the necessary OpenCV is contained. There are various guides to installing OpenCV 2.4 with Visual Studio.