Finding the Convex Hull with Application to Distance Measurements for a Formula Student Car

May 25, 2020

1 Introduction

Hi My name is Benedikt Schlereth and I'm studying Applied Math and Physics in the sixth Bachelorsemester. My Project is to find the convex hull to detected cones in a Image for measuring the distance. Right now I'm working for the Fraunhofer-Institut IIS here in Nuremberg, where we want to find the best feature-extractions tools to detect certain behaviors. To learn the practical implementation of new technology I support the local Formula-Student-Team Strohm und Soehne by designing PCB-Boards for our Electric car. In the future I also want to help out in our Software department.

2 motivation

First let me explain what Formula Student is: Formula-Student is an international design and construction competition where teams from all around the world are constructing and then racing single seat formula race-cars. With the new event of Formula-Student-Driver-less Teams are encouraged to develop their own self-Driving cars. My motivation is to kickoff the development of our own Driver-less Vehicle by implementing the first step and providing a way to detect the boundaries of a race track.

The car will drive on a set course marked with blue and yellow cones. So the only objects that the car needs to detect are the said cones and no other cars or people. There a strict safety-Rules in place, to protect man and machine which aren't of any interest for this project.

3 starting

A complete self-Driving car for such a competition would be a project for a few students, so I will focus only on detecting the edge of the track. To detect the boundaries a computer has to visualize the cones. My first step is to extract only the important colors from a camera mounted to the car to filter out the background and street. So with only the racecar on the track the only blue or yellow regions of the image should be the pins. Now the image should only consist of the cones that due to Image noise have odd shapes and no clear edges.

4 Algorithmus

To get a clearer Image with sharper edges to examine I can calculate the Convex hull of the remaining objects inside the Image. The Convex Hull is the smallest region containing all points while having no Concave corners. It is an important step for a few image-processing Algorithms, so development for so called Gift-Wrapping-Algorithms are dating back quit far in computervision-technology. To lose not to much time on calculation it is important to choose an fast algorithm. There are several available and one is already implemented in OpenCV, an Open Source Python-Library for computer-Vision. But there are several drawbacks with Gift-wrapping algorithms. Most of them are slow and the fast ones are proved to be wrong on some special occasions.

5 publication

And now lets have a look on the algorithm by Graham and Yao. The first step is to connect points in a given dataset with a straight line and walking around this dataset in a clockwise fashion. Every vertices is now marked with a start and end Location. Two vertices p and q having one Point in common can be written like p dot q. Now if we are looking at P dot Q and walking along those lines we can examine a third point Z. If Z lies on one of the lines or to the right of both vertices, the point is part of a so called pocket if it lies to the left of at least one vertices it is not part of the same pocket. Connecting more vertices to a closed loop gives us a pocket with some points already enclosed by one hull.

This has to be repeated until all points are enclosed by hulls. To find the convex Hull all the pockets or left hulls need to be connected to a big polygon containing every point. Therefore we choose the minimum and maximum Points of all left hulls and connecting those points. If two vertices are not convex then there common point will be rejected and left out in a way it will still stay in bound of the convex hull.

6 wrap-up

While being a not common known feature for shapes, the convex hull delivers good edges of objects. So with detecting the outlines of the cones I should be able to mark the track-boundaries for the car.