Team: Project 42

Requirement Analysis

Program to understand the movement of car from a target video

Team:	**U1610041**	**Azizbek Kobilov**
	U1610016	**Akmal Karimov**
	U1610047	**Bekhzodkhon Makhkamov**
	U1610052	**Boburjon Bahodirov**

Contents

1. Introduction

Purpose of introduction part, is to provide a brief overview of the function of the system and the reasons for its development, its scope, and references to the development context (e.g., reference to the problem statement written by the client, references to existing systems, feasibility studies).

1.1 Purpose of the system

The main purpose of the system is to build an image pipeline, which takes a frame from video as an input, do some calculation, and return modified frame. Of course, it is not the self-driving car itself but image pipelining is one of the key factors to implement that system.

1.2 Scope of the system

The program for understanding the movement of the car can be used in various large projects, for example, in the project of an autonomous car, as a module that helps the basic algorithm of an autonomous car. It uses streaming video from a camera located on the front of the car as an input and outputs the processed video with detected lane lines.

Moreover, the program can be expanded and used outside of autonomous automotive projects. Let's talk about one of his applications. If we place the camera and install the program on any car, it can stop the car or warn the driver if he falls asleep. In this case, if the driver falls asleep, his car will leave the road. This can be detected by analyzing lane lines.

1.3 Objectives and success criteria of the project

The main criterion to success is when car is traveling safely without any accidents. And the factor which leads to the first and main criterion is the CPU of the microcomputer which built inside the car, as computing and getting data will be real-time computer should be powerful enough to make needed calculations in time.

1.4 Definitions, acronyms, and abbreviations

Grayscale – technique where input image is in color but the output image is gray scale image.

ROI – region of interest, in computer vision, defines the borders of an object under consideration.

Hough lines – is a feature extraction technique used in image analysis, computer vision, and digital image processing

Threshold – is some fixed value which draws a boundary line between two set of data.

Binarization – is the process of converting a pixel image to a binary image.

Edge detection – the main purpose of this technique is to capture important events and changes in image.

Gaussian blurring – also known as Gaussian smoothing is the result of blurring by a Gaussian function, to remove the noise.

Canny edge detection – is an edge detection operator that uses a multi-storage algorithm to detect a wide range of edges in images.

1.5 References

https://www.youtube.com/watch?v=gWK9x5Xs_TI

https://towardsdatascience.com/carnd-project-1-lane-lines-detection-a-complete-pipeline-6b815037d02c#.hcsizymg8

https://www.youtube.com/watch?v=eLTLtUVuuy4

Main source:

https://www.udacity.com/course/self-driving-car-engineer-nanodegree--nd013

and repositories of this nanodegree program course.

Also, slides and online video materials provided in lectures.

1.6 Overview

Below parts describe some functional requirements, cutting, detection techniques applied etc.

2. Current system

Current system, describes the current state of affairs. If the new system will replace an existing system, this section describes the functionality and the problems of the current system. Otherwise, this section describes how the tasks supported by the new system are accomplished now.

3. Proposed system

Section documents the requirements elicitation and the analysis model of the new system.

3.1 Overview

The overview presents a functional overview of the system. The program analyses the streaming video from the camera located on the front of the car. It uses different image processing techniques to gain useful data. At the end, the user receives a video result with detected lane lines.

3.2 Functional requirements

Functional requirements describe the high-level functionality of the system.

3.2.1 Image Resizing

Frame processing should be started by resizing image into 1024x768, 800x600, 640x480, 400x300 dimensions.

3.2.2 Image Cutting

Frame needs to be cut horizontally in half, to remove redundant part and increase processing speed. Sky part of the image has been cut because we do not need it, this has been made for only purpose and it is for optimization.

3.2.3 Image Denoising

Program should denoise image using Gaussian blur technique. Reason for this, is to remove noise from image. Also, one of the advantages of chosen technique is speed.

3.2.4 Grayscale

Required to cast 3 channel image to 1 channel, taking its grayscale. Since we don’t need colors.

3.2.5 Edge Detection

Edges can be detected using Canny edge detection method, since it returns image with smooth edges. Blurred image is passed to Canny method.

3.2.6 Hough Lines Detection

Hough lines are detected using predictive method and all lines stored. Method expects image with detected edges as an input, and return all lines in an image. So, we pass output of Canny method to HoughLinesP method.

3.2.7 Separation of Lines

Output of method HoughLinesP is array of lines. Lines are divided into left and right, depending on their slope. Lines with a positive slope, to the right lines, lines with a negative slope, to the left.

We find the median of all lines on the left, the same we do for the right lines. This gives us two solid lines on both sides, after we return those lines.

3.2.8 Draw Lines on Initial video

Solid lines are drawn on an empty image of the same shape as the original frame.

3.2.9 Position Detection

Turn prediction is accomplished using slopes of both lines. We compare them to some fixed value, in the end, it tells with medium accuracy, where car is moving.