TOSHKENT SHAHRIDAGI INHA UNIVERSITETI
INHA UNIVERSITY IN TASHKENT

Computer Science and Software Engineering

Comprehensive Assignment: Software Design

Capstone Design

TEAM NAME: DEUS EX MACHINA

TEAM NUMBER: 50

TEAM MEMBERS:

1. JAMOLIDDINKHUJA ODILKHUJAEV: U1610092
2. KAMOLA AZIMOVA: U1610101
3. MAHLIYOKHON OLIMJONOVA: U1610133
4. MOKHLAROYIM TUYCHIBOEVA: U1610148

Detail SW design

Assumptions and Dependencies

The first instinct to us, humans, as drivers, is probably to look in front of us and decide where should the car move; at which direction, between which lines, etc. As every Autonomous Vehicle comes with a camera in a front, one very important task its to decide the border in which the car should move in between. Here we impose an assumption that cars should follow the certain logical or structural sequences of lanes.

Dependencies: We will be leveraging the popular SciPy and NumPy packages for doing scientific computations and the OpenCV package for computer vision algorithms

Goals & Guidelines

We will develop a simple pipeline using OpenCV and Python for finding lane lines in an image, then apply this pipeline to a full video feed.

SYSTEM FEATURES

1. Image denoising
   - is the task of removing noise from an image, e.g. the application of Gaussian noise to an image? Gaussian filtering is done by convolving each point in the input array with a Gaussian kernel and then summing them all to produce the output array.
2. Edge detection from binary image
   - It is a multi-stage algorithm and we will go through each stages, the theme of which will be revealed in the sector below
3. Mask the image
   - The idea is that we recalculate each pixel's value in an image according to a mask matrix (also known as kernel). This mask holds values that will adjust how much influence neighboring pixels (and the current pixel) have on the new pixel value. From a mathematical point of view we make a weighted average, with our specified values.
4. Hough lines detection
   - The Hough Line Transform is a transform used to detect straight lines.
   - To apply the Transform, first an edge detection pre-processing is desirable.
   - We are going to use probabilistic Hough lines in OpenCV to identify the location of lane lines in the road.
5. Left and right lines separation
   - Being able to detect lane lines is a critical task for any self-driving autonomous vehicle.
   - The blur and grayscale step will help make the main lane lines stand out.
6. Drawing the complete line
7. Predict the turn

SYSTEM ARCHITECTURE

First of all, we want to make the image into a grayscale one; only one color channel. This will help us with the identification of edges and corners.

Adding Gaussian noise to an image, it very useful as it smooths the interpolation between the pixels and is a way to super-pass noise and spurious gradients. Higher the kernel, the more blur the outcome image will be.

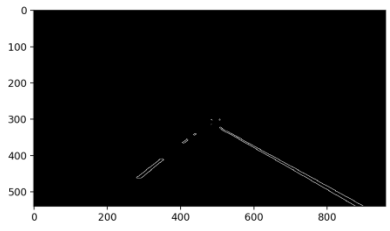
Canny Edge Detection offers a way to detect the boundaries of an image. This is done through the gradients of the image.

The latter is nothing more than a function, where the brightness of each pixel corresponds to the strength of the gradient.

We will find the edges by tracing the pixels that follow the strongest gradients! As in general the gradients show how rapidly a function changes, an intense density change between the pixels will indicate an edge.

After this step we will can clearly identify the road lines.

But on the picture, will be some outliers; some edges from the other part of the road, from the landscape (mountains), etc. As our camera will be fixed, we can put a mask upon the image and keep only these lines that are interesting for our task. It's important to cut out as much of the noise as possible within a frame. Given the position and orientation of the camera, we can assume that lanes will be located in the lower half of the image. We can just isolate that area using a trapezoid shape. While it may be tempting to hardcode the best possible place to look for lines, lines can move around as you drive. For this reason, a simple ratio to setup a polygonial shape is used. We apply a generous ratio as we don't want our region to be too narrow resulting in missing lanes out of the region of interest.



Using probabilistic Hough lines, we identify the location of lane lines on the road. The Hough transform algorithm extracts all the lines passing through each of our edge points and group them by similarity. The above image represents only the dots of the edges. What is left is to connect the edges. In this case, we are looking for lines and we are going to do this by transporting the images to the parameter space, called *Hough Space*. We will now deal with *polar coordinates* (rho and theta), in which we will search for intersecting lines.

We will have many small lanes, after above step. The next step will be connect these lines and the result will be only two, that they will be road ones.

Our strategy will be the following:

- Break the image in half with reference to the x-axis

• Fit a linear regression model to the points, in order to find a smooth line.

Due to outliers, we want a regression model that can deal with them efficiently. We are going to use the HuberRegressor. Then, we will bound the image between a certain range of the y-axis and with the help of cv2.polylines we will plot the line. As a result we will get two lines.

Next, we will concatenate the lines with the original image. By giving weight to the two images, we can add them. All process will be shown on section Methodology with bright illustration our coding part.

References

OpenCV. (n.d.). *https://opencv.org/about/*.

- https://www.researchgate.net/publication/323406613_VEHICLE_NAVIGATION_USING_ADVANCED_OPEN_SOURCE_COMPUTER_VISION
- Jay Hoon Jung, Yousun Shin, YoungMin Kwon (2019). "Extension of Convolutional Neural Network with General Image Processing Kernels".
- Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (2016). "You Only Look Once: Unified, Real-Time Object Detection".
- Mayank Singh Chauhan, Arshdeep Singh, Mansi Khemka, Arneish Prateek, Rijurekha Sen (2019). "Embedded CNN based vehicle classification and counting in non-laned road traffic".
- Tejas Mahale, Chaoran Chen, Wenhui Zhang (2018). "End to End Video Segmentation for Driving: Lane Detection for Autonomous Car".