Lane-Detection: CSCI-431 Final Project Proposal

Evan Hirsh and Aidan Rubenstein
Rochester Institute of Technology

ABSTRACT

The process of using lane detection within the scope of Computer Vision (CV) for the assisted driving and auto-pilot technologies in automobiles has been a challenge that still stumps automotive manufacturers and researchers in the modern day. In essence, most modern lane detection techniques use camera technologies that are able to get multiple views of the surrounding area of a car, in conjunction with a convolutional neural network in order to identify possible objects or obstructions. However, due to the cost of having to strategically place 4-8 cameras around a vehicle and the lack of affordable technology (such as LIDAR devices to detect distance) to allow the car to take driver actions, this type of CV solution is not viable for older models of vehicles. The goal of this paper is to incorporate key CV algorithms in order to create a basic lane detector that can be compatible with any smartphone or device that is capable of live-streaming/recording video. With a sample video, the edges of each individual frame can be determined in both the x-direction and y-direction by means of the Canny Edge detection process, and then be segmented into a region of interest (ROI) that eliminates possible false-positive values on the road. Lastly, the Hough Transform algorithm is applied to find the lane gradients in the ROI and will be indicated on the original video feed to highlight the detected lane.

KEYWORDS

autopilot, assisted-driving, computer vision, lane-detection

ACM Reference format:
Evan Hirsh and Aidan Rubenstein. 2020. Lane-Detection: CSCI-431 Final Project Proposal. In *Proceedings of NA, NA, 2020,* 3 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Over the course of the past century, the invention and revolution of the automobile has been the most important technology in the modern world. As more and more vehicles become available to the average consumer through consistent manufacturing and depreciation of car values, the rate of driver-related accidents have increased [1]. In recent decades, innovations have been discovered in an attempt to make driving more safe. Common driving features in newer cars include lane-departure warnings, which will indicate to the vehicle operator if he/she is drifting from the lane that they are in, and adaptive cruise control in order to reduce the risk of collisions and fatalities in the vehicle.

In most commercial vehicles, they way that this technology is implement is mainly through the use of cameras, and communication between these peripherals and the main CAM bus that is essentially the car's "motherboard". These cameras are usually place at the front and rear sections of the vehicle, as well as one camera for both the left and right blind spots. In the case of Tesla's latest autopilot feature, which relinquishes full vehicle control to the CAM bus at the user's discretion, it uses: a forward looking radar, camera, and 12 long-rage ultrasonic sensors placed around the car to give full 360 degree coverage and a maxium distance of 16 feet in all directions [5]. However the issue with these technologies is the relative cost to implement them on common automobile makes and models that have been in circulation before these driving technologies became accessible.

In the field of Computer Vision, feature detection is one of the core concepts around image manipulation, and algorithms have been developed and improved over decades in order to increase the effectiveness of these processes even before the era of driver-assistance. The ability to use edge detection to identify curves and lines within an image is a pragmatic solution to many of the CV problems that exist, such as: face detection, neural networks, and improving classifications of subjects. The presence of lanes as a guiding tool for drivers on the road is simply another indication of a feature within an overall image, and as such common CV techniques can be used to find these lanes on a system that can be accurate yet affordable for the average consumer.

In this paper, we provide a suggestion to incorporate these lane detecting mechanisms in more common vehicles by being able to process all of the frames of a video feed for lanes on a standard United States highway. Once a viable image of the subject matter has been captured, a Canny Edge detection algorithm will be able to detect the magnitude and direction of possible edge candidates in the horizontal and vertical directions. From there, we create a custom ROI that filters out possible false-positive edges with gradients, such as guardrails or other larger vehicles. Due to the nature of the perspective and angle the driver takes when operating a vehicle, this ROI can be shaped into a natural triangle/cone formation, as the lanes will eventually touch together at a point ahead of the driver, indicating depth. Lastly, the Hough Transform algorithm will be used to determine what edges in this ROI follow a solid gradient, which should ideally be the target lanes that we are seeking to find. Once the lanes have been detected and their positions found relative to the original frame, the pixels of these lanes are recolored for a visualization of the final result of the lane-detection process.

2 PREVIOUS RESEARCH

Since the lane-detection feature has already been observed as being present in other technological implementations in newer cars within the decade, it is important to note what research has been done in the way of progress for solving the lane-detection problem

NA, NA

DOI: 10.1145/nnnnnnn.nnnnnnn

as a whole. In prior research, the accuracy of lane detection has been improved by detecting and seperating the edge gradients and directions for the horizontal and vertical planes. The horizontal plane is combined with the Otsu's Threshold method in order to identify all lanes in front of and adjacent to the vehicle, and any obstacles that are within range of the vehicle. In the vertical direction, the edges are used to verify the horizontal lane/vehicle candidates, and the Kalman filter is applied in order to track the direction of possible approaching vehicles [4].

Other research has pointed to the idea that color space in which the image of the road is viewed also has an impact on the accuracy of the detection algorithm. Researchers have determined that converting an image in the traditional RGB color space to the HSV color space has better results in terms of accuracy in the system. Once converted, the whole image is combined with the Kalman filter to give a final result of the lane-detection process [2]. However, in both research observations, the only element that is missing is the speed that is required by lane-detecting algorithms in order to make quick decisions and alert the driver if an imminent danger is present, which will be the objective of this paper.

3 IMPLEMENTATION

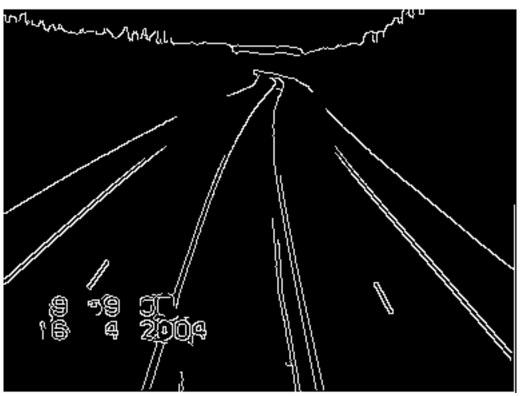

Figure 1: An image of a road processed by a Canny Edge Detector [3]

Our solution will be a service implemented in Python 3.7+ that will utilize implementations of Canny Edge Detection and Hough Transform from the common mathematics and image processing libraries OpenCV and NumPy to process images polled from conventional sensors. We first poll the sensors at a specific interval of which will be determined by the speed of the processor being utilized for the detection. Once that image has been loaded into memory, we begin the first phase of the implementation, the canny edge detection. For the purpose of this project, the RGB color space will be used which will provide a less accurate, however computationally effective algorithm. Using a Gaussian filter, the image is smoothed in order to reduce the noise to prevent outliers from impacting the later steps in the process. Next, we apply the Sobel gradient filter to detect the edges in the horizontal, and vertical directions of the image, and then calculating the overall gradient and angle using the Pythagorean Theorem. Non-maximum suppression is then utilized to help "thin" the edges in the image, suppressing all the values in the gradient that are not the local maxima. The double threshold is then applied, removing pixels with weak gradient values that are caused by noise and color variation. The final step in Canny, hysteresis is then used to track the edges and remove "weak" pixels. The resulting image should display outlines indicating the edges of the lane indicators in the road.

Once the edges in the image have been detected, we must now triangulate the region of interest in the image (the lane markings) to eliminate the false positives of the other edges. This will be achieved by averaging out possible areas within the image frame as to where the lane would begin and end, and create the third point of the triangle around half of the height of the image dimensions. This triangle essentially acts as a image "window" with the triangle being the value 255 and all other values set to 0, which will eliminate all outside portions of the image by using a bitwise-and operator with the black and white mask. Once that is completed, we will use a Hough Transform on the matched areas to identify the slope-intercept values at every pixel in the space, which can be averaged out to find the best-fit line for where the lane would be. Once the avearge slop-intercept form is decided, we can plug every pixel location back into the formula to find where the lane will begin. Lastly, we will use those calculated vectors to draw our results back on the original image for testing purposes.

4 RESULTS / FINDINGS

Figure 2: The sample image utilized for testing and filter creation

Figure 3: lane$_{d}etector.pydetectingthelocationoflanesonanormalPNGfile$

Within the time limits allocated and with the restrictions we placed on ourselves, we were able to implement a working system that allowed us to detect the lane from a normal camera sensor. Our implementation is a single file multithreaded python solution which has the ability to take an image or video file, and extrapolate the position of the lanes using the methodology described previously. In the case of a video file, it does this by splitting each frame into a single thread, processing them separately. It utilizes standardized python libraries such as NumPy and OpenCV, and the final submission has slightly over 250 lines of python code excluding the aforementioned dependencies.

As we expected initially, there were some issues we encountered with the program based on the nature of the data we are trying to infer, and the limitations of the sensors themselves. We discovered that the standard convolution we are utilizing with the Sobel edge filters was computationally inefficient for the speed at which the information would need to be provided to the automobile's systems. While it is not expected that lane data should be extrapolated from every frame polled by our sensor, our operations are taking upwards of ten seconds to compute, which makes it difficult to operate on CPU's with slower clock speeds. This can be improved upon in future versions through either linear separation of the filters, or memoizing the convolution operations so that matrix operations do not need to be performed. Additionally, the limitation of a single camera poses an interesting challenge for filter creation. While we were able to properly detect the lanes on the sample image, we were able to use a filter that we built based off the height and orientation of the sample image provided. This means that our program becomes less accurate the further the dimensions, position, and orientation of the sensor are from the one that we utilized, and numerous additional filters would need to be created to ensure accuracy of all sensor types.

REFERENCES

[1] National Safety Counsel. 2018. Car Crash Deaths and Rates. (2018). https://injuryfacts.nsc.org/motor-vehicle/historical-fatality-trends/deaths-and-rates/

[2] Mingfa Li, Yuanyuan Li, and Min Jiang. 2018. Lane Detection Based on Connection of Various Feature Extraction Methods. *Advances in Multimedia* 2018 (Jul 2018), 1–13. DOI:http://dx.doi.org/10.1155/2018/8320207

[3] Cristina Maduro, Katherine Batista, Paulo Peixoto, and Jorge Batista. 2008. Estimating Vehicle Velocity Using Rectified Images. (01 2008), 551–558.

[4] VanQuang Nguyen, Heungsuk Kim, SeoChang Jun, and Kwangsuck Boo. 2018. A Study on Real-Time Detection Method of Lane and Vehicle for Lane Change Assistant System Using Vision System on Highway. *Engineering Science and Technology, an International Journal* 21, 5 (2018), 822 – 833. DOI:http://dx.doi.org/https://doi.org/10.1016/j.jestch.2018.06.006

[5] Cadie Thompson. 2016. Here's how Tesla's Autopilot works. (Jul 2016). https://www.businessinsider.com/how-teslas-autopilot-works-2016-7