Enhanced Lane Detection System with Image Pre-Processing

Himath Ratnayake

Introduction

Lane detection is a mechanism that is currently used in higher end cars as a technique that assists drivers in identifying lane markings. There are a variety of practical applications in which lane detection proves useful. This ranges from lane departure warning systems, which aim to assist drivers in keeping center within their lane, or to warn drivers when they veer off the lane in a dangerous way. Cars that employ this system have shown to visibly improve driver safety by minimising driver error resulting from distractions or drowsiness. More recently, lane detection systems have earned a place as an integral subsystem of self-driving cars. However, there are still several challenges posed by current lane detection systems that hinder the efficacy of autonomous driving. There are many common driving situations in which issues can still occur; this includes night-time driving when visibility of lane markers is unclear, as well as when other road markings such as brake marks or shadows hinder the vision of the input camera. Moreover, one of the biggest challenges effective lane detection systems have is the need to overcome noise appearing in the input image or video frames

Technical Approach

To tackle these issues, a number of image pre-processing tasks will be carried out beforehand. First, an input frame from a video is read in. Afterwards, the incoming image will be cropped so that only a specific region of interest surrounding the immediate lane markers in front is present. In this way, unwanted foreground features that may be picked up such as car headlights are reduced. There is also a computational advantage to be had by working with a smaller image, as there is less pixels with which to carry out Canny and Hough Transform computations. Then, the input frame is converted to grayscale. Converting an image to greyscale aids in reducing image complexity while still being able to highlight key image features such as shape, contrast, perspective and edges.

Finally, a filter is applied to reduce noise within the image. There are two main options in filtering that is used for denoising an image. The standard option that is generally used is a Gaussian blur filter. However, in this report, the use of a bilateral filter will also be explored to compare and contrast how it fares within high-noise environments like night-time driving as opposed to a normal Gaussian filter. In Gaussian filtering, the filter is applied uniformly. However, in bilateral filtering the similarity between the central pixel where the filter is applied and a pixel in its neighbourhood used for blurring is computed. In doing so, bilateral filtering has the effect that if the two pixels compared are very similar, the gaussian filtering will happen as normal. However, if the pixel values are very different, the filtering will not be done, thus skipping over the gaussian filtering for the given pixel.

Then, the Canny edge detection process begins. Within the open cv library, this can be done with the Canny() function. The idea of Canny edge detection is to identify points of interest in the image where the image brightness changes drastically; these places where a drastic change occurs are termed as edges.

After this, the next step of Canny edge is to find the intensity gradients of the image, and then apply gradient magnitude thresholding and non-maximum suppression to reduce spurious edge responses.

Canny edge detection uses gradients for the measurement of pixel intensities in output edges. A line of best fit can then be found for each set of these gradient points. These lines of best fit will ultimately become the lane lines. To achieve this, Hough Line Transform will be used, with the cv2.HoughLinesP() function from the open cv library. With this, the final lines of best fit will be overlayed on the processed image and outputted. This full process is outlined in figure 1 below.

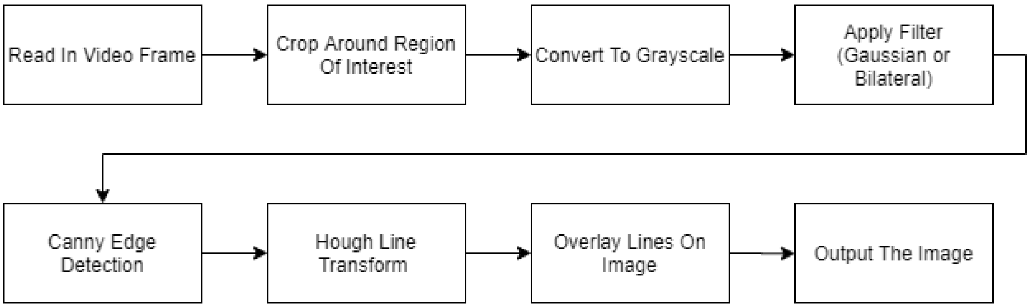

Figure 1: Flowchart describing the computer vision techniques undertaken to detect the lanes on the road

Expected Outcome

The expected outcome is that this algorithm will be able to effectively detect the presence of lanes within a variety of settings, including ones where typical lane detection algorithms may struggle to differentiate instances of noise.

Bilateral filtering is also expected to aid in more effective noise reduction, when compared to a normal Gaussian filter. This is because of the fact that – as defined earlier – filtering effects aren't uniform like a Gaussian filter, and the filtering effect isn't as prevalent in areas where pixel neighbourhoods have drastic differences. This means that edges, which are essentially defined as areas of drastic change in pixel intensity, are expected to be better preserved in bilateral filtering as opposed to Gaussian filtering. It will remove noise in uniform areas like a Gaussian filter, but not introduce blur between objects.

In terms of the output, a typical example of a lane detection system's output is shown below in Figure 2.

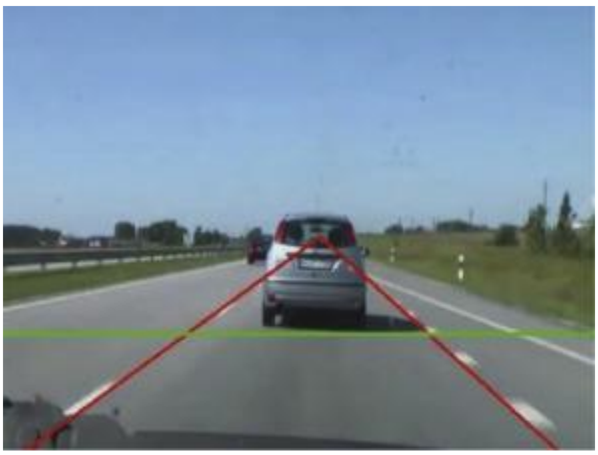

Figure 2: Typical output of a lane detection system (Ferdinand, N., 2020)

Key Milestones

The key project milestones have been illustrated below in Table 1.

Table 1: Key Milestones and Project Goals		
Date	**Milestone**	**Notes**
September 10th	Complete project proposal draft	Finish a rough draft of all sections and compile references
By September 13th	Proofread and complete project proposal	Complete references section after proofreading is finished
September 14th	Set up project report and create skeletal structure of coding environment	Set up Jupyter notebook on Python, as well as outline key functions/libraries necessary for the program
September 19th	Submit project proposal on L@G	
By September 26th	Complete coding of the project on Python	Subgoals include implementation of: • Frame pre-processing • Canny • Hough transform • Image Output
By October 5th	Complete the project report	Subgoals of project include: • Title and Author • Abstract • Introduction • Previous Work • Technical Approach • Experiments • Conclusions • References
October 6th to 10th	Drafting Period	Ensure the project code and report is well formatted with references, under the page limit, and that all code is ready for demonstration.
October 10th	Submit full report on L@G with code	
By October 13th	Prepare code demonstration and video presentation	

References

- Paris, S., 2007. A Gentle Introduction to Bilateral Filtering and its Applications. MIT CSail.
- Sultana, S. and Ahmed, B., 2021. Robust Nighttime Road Lane Line Detection using Bilateral Filter and SAGC under Challenging Conditions. 2021 IEEE 13th International Conference on Computer Research and Development (ICCRD),.
- Liu, D., Wang, Y., Chen, T. and Matson, E., 2020. Accurate Lane Detection for Self-Driving Cars: An Approach Based on Color Filter Adjustment and K-Means Clustering Filter. International Journal of Semantic Computing, 14(01), pp.153-168.
- Ferdinand, N., 2020. A Deep Dive into Lane Detection with Hough Transform. [online] Medium. Available at: <https://towardsdatascience.com/a-deep-dive-into-lane-detection-with-hough-transform-8f90fdd1322f> [Accessed 9 September 2021].