Multiview Geometry in Computer Vision: Lane Recognition and segmentation on left-hand-drive Irish roads.

Senan Stanley

March, 2023

Abstract

Lane detection, recognition and segmentation are pivotal problems in the automotive industry as mainstream autonomous vehicles (AV) teeter on the precipice of market penetration. This paper intends to introduce a lane detection method based on classical computer vision techniques which performs robustly on a variety of lane-hand drive Irish roads. Data is extracted from a video stream frame-by-frame, filtered and processed to find likely lane markings and graphically represents these markings on the original frames.

Keywords: Machine Vision, Autonomous Vehicles, Lane Detection, Road Segmentation, Irish Roads

1 Introduction and Background

Lane detection is used to gauge the immediate position of an ego vehicle on a road or track. This information is important to ensure the vehicle is legally occupying a roadway. As well as this, the knowledge of the lane geometry gives insight to an AV on future road topography, as well as isolating an area of interest (AOI) for further analysis. Aside from autonomous robotics, lane detection has also proven effective in human-driven vehicles, to check for anomalies in the road ahead, or deviations from the safest path [Yenikaya et al., 2013].

This technology is plainly important, it is the foundation for more complex AV undertakings of localisation, mapping and path planning. Therefore, it is the intent of this paper to outline a robust method for mono-scopic lane detection, validated on various Irish roads.

The field of left-hand drive lane detection algorithms are at a disadvantage as only 30% of the world's countries drive on the left-hand side of the road [World-Standards, 2023]. Occlusion of the road by weather conditions or other vehicles also inhibits the lane recognition process. As well as this, some roads are poorly maintained, with no road markings visible.

Much of the ongoing research in lane recognition is being attempted using neural networks (NN). The NN approach lends itself to a better framework of metric assessment, as supervised learning models require ground truths for comparison and thus can be easily quantitatively analysed. For experimental algorithms, such as this, the efficacy of the results can often only be judged qualitatively, based on conditions and performance [Dinakaran et al., 2021, Sharma et al., 2021].

2 Method and Results

All implementation of lane detection was completed using the OpenCV and Python3 Programming language, run on Google Colab platforms for GPU deployment [Bradski, 2000, Van Rossum and Drake, 2009].

Lane detection was completed in two steps; pre-processing of the input image & lane recognition using the processed image. Though the original data used was in video form, each frame of the video was processed one by one to detect lanes throughout the file.

2.1 Pre-processing

Each input image was first cropped into a trapezoidal AOI, the dimensions of which are based on the measurements of the image and coefficients calculated from sampling data. The intent of this step is that the trapezoid contains all pertinent information for the road and road markings, while discarding excess information.

The trapezoidal area was then converted to greyscale values, via the averaging of each Blue-Green-Red (BGR) channels. This was further augmented using the Sobel–Feldman operator, in order to detect edges and ignore boundary-less areas [Sobel, 2014].

Simultaneously, the BGR image was converted to a Hue-Luminance-Saturation (HSL) image representation. The luminance (brightness) portion of the image was band-thresholded to isolate bright white and yellow values.

The Sobel filtered image and luminance filtered image were combined using a bitwise logical conjunction (AND) operation [Hazewinkel and Plisko, 1994]. The resulting image should isolate yellow and white edges.

The image was then blurred, to merge near pixels into larger structures, and thresholded into a binary image. Two cycles of open morphological operation were then employed. This involves the erosion and dilation of all pixels in a binary image, resulting in a denoised image, with fewer small, unconnected "islands" of white pixels [Haralick and Shapiro, 2000].

The final step of pre-processing was implementing the Canny edge detector, with a 3×3 aperture [Canny, 1986]. A graphical representation of the image pre-processing pipeline is displayed in Figure 1, below.

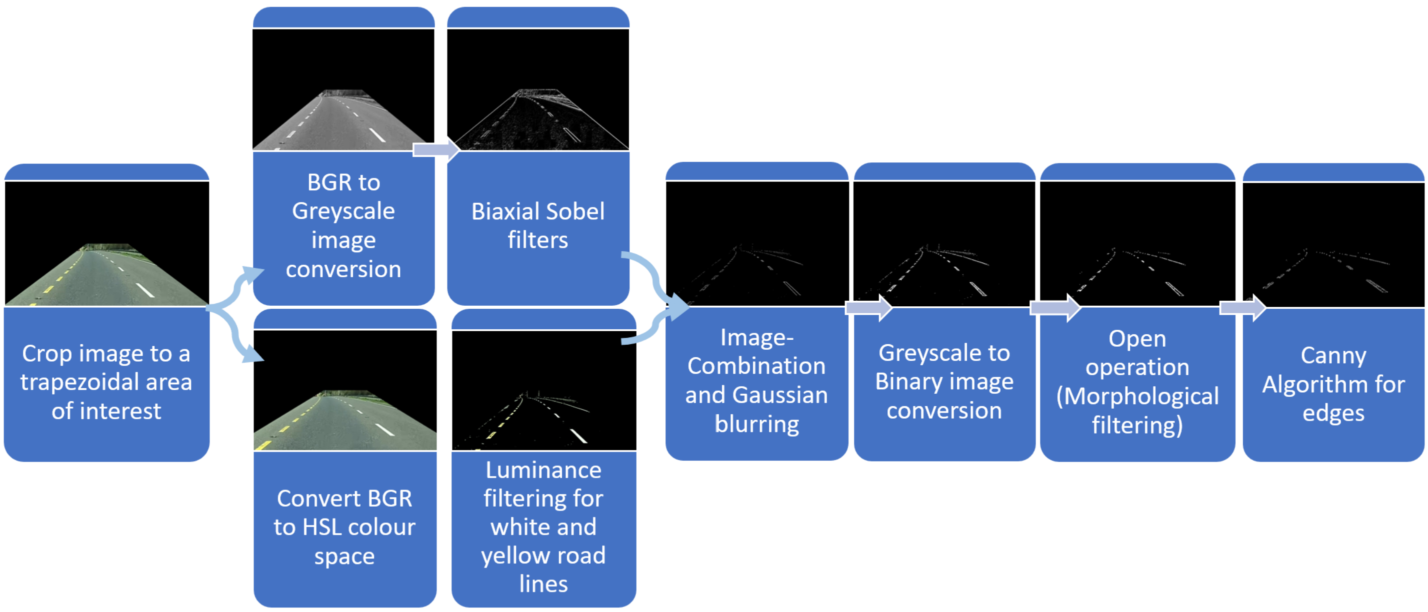


Figure 1: The pre-processing pipeline of each input image.

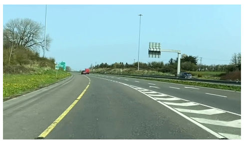
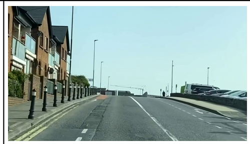
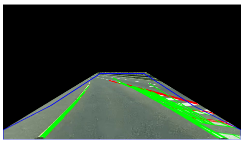
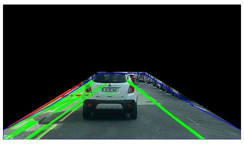
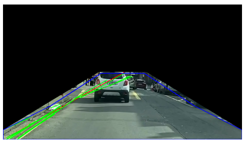
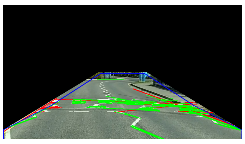
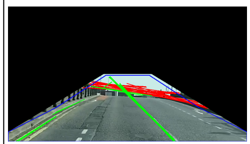
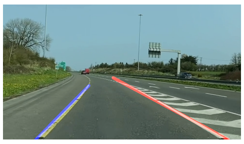
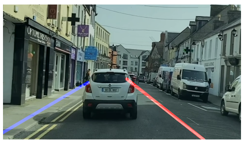
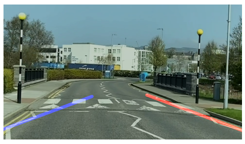
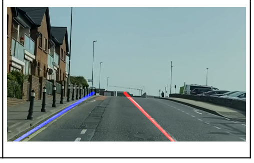
2.2 Lane Detection

After pre-processing is complete, the detection algorithm must be employed. The Hough Transform for lines algorithm was implemented to find continuous lines in the pre-processed image [Duda and Hart, 1972]. The intent of the pre-processing was to reduce the non-lane marking objects in the image, so that Hough lines could only detect the lane.

With the list of probable line coordinates detected in the image, the lane markings must now be recognised. Another polygonal AOI was created within the existing polygonal aperture, to assist in line filtering. The expectation is that each line's boundary coordinate must be contained within the smaller AOI to be a valid road marking. Thus, lines beginning in the extremities of the image are discarded.

Another assumption was made, that lines with slopes of extreme magnitude were deemed as non-road markings. Similarly, lines with constituent points too disparate from each other or below a minimum length were ignored.

Table 1: Comparison of 5 road scenarios and the predicted lanes

State	Motorway	Town	No Road Markings	Zebra Crossing	Regional Road
Original Image					
Hough Lines					
Lanes Marked					

The remaining connected lines were combined into two categories to constitute the left and right lane markings, this process can be seen in Figure 2.

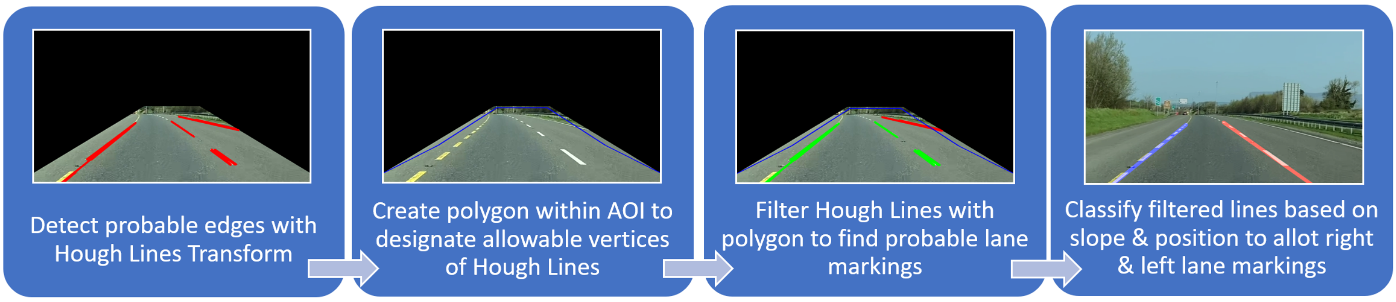


Figure 2: The lane recognition pipeline.

2.3 Results

This algorithm was tested using footage of differing lane conditions, these were: motorway, congested town road, town road without road markings, zebra crossing and single lane regional road. The algorithm was assessed on the error between predicted and true right and left lane positions. The original, detection and recognition states are compared in Table 1.

With fewer cars on the road to occlude markings and wider roadways, the regional and motorway roads had the lowest error rate of lane position. The town lanes were reasonably well predicted, with the algorithm incorrectly predicting some paths or shopfronts as lanes.

The lane predictions for roadways without markings were very often incorrect. Without the lane markings present in frame, the algorithm could not guess where the lane was. Similarly, an overload of lane markings can be seen in the Zebra crossing scenario. The algorithm overpredicted the lanes and struggled to sift through all the lines the Hough transform produced.

In an attempt to improve lower error, hyperparameters for all functions used were tuned. Trapezoidal areas were varied, thresholding values were tuned for optimal line coherence & kernels were altered for better edge detection and filtering.

3 Discussion and Conclusion

Many of the failures exposed by the results were due to the algorithm's inability to see perfect lane markings. An iteration of this algorithm must improve on the ability to take noisy data (faded/occluded markings) and forecast where lanes should be, based on prior knowledge. Also improving the algorithm's ability to take contextual cues of road width and non-ego vehicle position to estimate lanes.

Any areas of interest, such as the trapezoidal road assumption used in this paper, should be assigned dynamically, for this system to be as robust as possible. This would also allow for the camera to be resituated around the vehicle while retaining full functionality.

Further iterations can be built upon the current OpenCV Python implementation [Github:SenanS, 2023] and could include more features, to increase robustness and accuracy of lane detection. Features such as; better short term memory caches for following continuous lanes through time and a greater ability to predict the location of lane markings in more adverse conditions.

Though this algorithm is obviously immature and facile compared to the state-of-the-art research, the results here show not only what can be accomplished with a limited algorithm and a single optical sensor, but what infrastructure needs to be improved to help ease Ireland's transition to autonomous transportation.

References

[Bradski, 2000] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

[Dinakaran et al., 2021] Dinakaran, K., Stephen Sagayaraj, A., Kabilesh, S., Mani, T., Anandkumar, A., and Chandrasekaran, G. (2021). Advanced lane detection technique for structural highway based on computer vision algorithm. *Materials Today: Proceedings*, 45:2073–2081. International Conference on Advances in Materials Research - 2019.

[Duda and Hart, 1972] Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15.

[Github:SenanS, 2023] Github:SenanS (2023). Machine vision lane detection using opencv.

[Haralick and Shapiro, 2000] Haralick, R. M. and Shapiro, L. G. (2000). *Chapter 5: Filtering and Enhancing Images*, page 150–156. Addison-Wesley.

[Hazewinkel and Plisko, 1994] Hazewinkel, M. and Plisko, V. (1994). *Conjunction-Logical Operations*, volume 10. CWI.

[Sharma et al., 2021] Sharma, A., Kumar, M., and Kumar, R. (2021). Lane detection using python. In *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pages 88–90.

[Sobel, 2014] Sobel, I. (2014). An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*.

[Van Rossum and Drake, 2009] Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

[World-Standards, 2023] World-Standards (2023). List of left- & right-driving countries.

[Yenikaya et al., 2013] Yenikaya, S., Yenikaya, G., and Düven, E. (2013). Keeping the vehicle on the road: A survey on on-road lane detection systems. *ACM Comput. Surv.*, 46(1).