Low Budget Lane Detection Using B-Spline and Canny/Hough Estimation of Vanishing Points

Kaixin Chen
Stanford University
Stanford, CA
kaixinc@stanford.edu

Shunyao Xu
Stanford University
Stanford, CA
shunyaox@stanford.edu

Abstract

As autonomous vehicles are in rapid development, lane detection has been a hot topic over the past decade. In this paper, we first implemented a basic lane detection algorithm using Canny Edge Detector and Hough Transform. Based on this method, we implemented an improved algorithm presented by the closed source project [1], with a variety of improvements to [1]. The improved method uses B-Spline and Canny/Hough Estimation of vanishing points to estimate the road lanes. Compared to the algorithm that only uses Hough transform, the fact that the improved method can also work with curved roads enables it to handle more real-world scenarios. Moreover, compared to a deep learning approach for lane detection, our presented improved method can significantly reduce the cost by eliminating the need for training as well as any labels for the data. Given a single image input, it can generate results that are robust to shadow and occlusion. The implementation of both methods was started from scratch, and we also eliminated the use of external libraries like OpenCV, for example, we implemented our own lower-level functions like Canny, Hough and RANSAC.

1. Introduction

Autonomous driving is an emerging new technology that can bring significant changes to transportation. Meanwhile, it is also a field that heavily relies on computer vision. According to SAE's standard of autonomous driving, one of the important features of autonomous vehicles is autonomous horizontal control, namely, lane-keeping assistance systems. Identification of the lane is a perfect example of a real-life application of computer vision.

In this paper, we proposed two approaches for lane detection. The first approach is a basic algorithm that involves Canny Edge Detector and Hough Transform. This approach is implemented only because it provides us with the tools for the second approach. We use Canny Edge Detector to extract edge features from the original image and then use Hough Transform to detect lines in the edge map. However, this method is restricted to straight lanes, while road lanes could be curved in many real-world scenarios. To improve the generalizability of autonomous driving, we implemented a second approach based on the first one. This approach is originally presented by [1], while we made a variety of improvements. Though this is a reimplementation with modification, this paper did not consult the code from [1] as it is not available. In autonomous vehicle applications, the actual position of the lane line is not used to navigate the vehicle. Instead, a virtual centerline of the lane is what guides the vehicle's steering. Therefore, computing a representation to the center of the lane is the focus of this method. The algorithm first segments the input image into horizontal stripes and computes the vanishing points corresponding to the lane lines in each of the stripes using the Canny edge detector, Hough transform and RANSAC. Then, these vanishing points become good indications of the curvature of the road, which will be used to give an estimate of the knots for a B-spline. Eventually, the B-spline captures the curvature of the road and detects the lane.

2. Problem Statement and Related Work

2.1 Problem Statement

We expect our lane detector to have the following properties:

1) *Robust to disturbance, which includes poor lighting and different line types.*

In a real-world driving scenario, it's very likely for an autonomous vehicle to encounter disturbance, including poor lighting and shadows. Therefore, we expect our algorithm to be able to detect lanes accurately even under these scenarios.

2) *Handles curved roads instead of assuming the lanes are straight.*

As mentioned in the previous section, our basic approach is restricted to detecting straight lanes, while real road lanes are very likely to be curved. We expect our approach to be a robust detector that can recognize both straight and curved roads. Also, a good method should be able to recognize yellow and white, solid and broken lines.

3) *Do not require a large number of labels and datasets or computationally intensive training.*

Usually, algorithms with robust outputs rely on vast training data, which prevents students, companies, and research groups with low budgets to work with such models. As such, for this paper, we do not expect our lane detector to have a computationally intensive training process that requires a large-scale labeled dataset.

The dataset used in this paper is obtained from [2]. Since this dataset contains a large number of images with lane markers taken from a vehicle's perspective view, it will be more than sufficient for tunning the parameters in Canny edge detection and Hough transform as well as providing the images for test cases. As such, this paper only uses a minor portion of the dataset in [2].

2.2 Background/ Related Work

Existing work in lane detection can be categorized into three main categories of approaches.

Firstly, is a learning-based approach. A learning-based approach can potentially achieve a state-of-the-art performance with good generalizability and high robustness. [2] is an example of learning-based approaches. However, such an approach requires a vast dataset and ground truth labels. For example, in [2], the dataset includes 100,000 images with 3d lines, pixel-level dashed markers, and curves for individual lines, and the entire dataset has a size over 100 Gb. For student and research groups running on a low budget, it is almost impossible to collect such an original dataset. In addition, the computing power needed to process the dataset is enormous. [2] used a GTX 1080Ti GPU, one of the most high-end GPU at the time. However, it cannot be assumed that such computing power is accessible by everyone. For example, the computing power we have access to is only an RTX 2060 Max-Q laptop GPU, which has much lower performance and smaller CUDA memory.

The second category is feature-based approaches. These approaches detect the location of lanes by combining the low-level features, for example, lane edges. However, such techniques usually require clear lane edges in the image. Also, because there are no constraints on the lane edge shapes, occlusion and noise severely impact their robustness.

The last category is a model-based approach, this category includes our proposed method. Model-based approaches use a few parameters to represent the lane, for example, the knots for B-spline in this paper. This category assumes the shape of the lane as straight lines, parabolic curves or, in this paper, spline. Model-based approaches generally show better performance than feature-based approaches. Though learning-based approaches usually outperform model-based approaches, the focus of the paper is a low-budget algorithm for lane detection, which made model-based approaches still provide state-of-the-art generalizability and robustness.

3. Technical Approach

3.1 Basic Method

The basic approach involves only Canny Edge Detector and Hough Transform. As shown in Figure 1, a Canny Edge Detector performs the following steps to extract the edges:

1) *Smoothing the original image using a Gaussian kernel.*
2) *Compute gradient in x and y direction.*
3) *Apply Non-Maximum Suppression to convert thick and blurry edges into sharp edges.*
4) *Use double-thresholding algorithm to filter out edges due to noise and color variation.*
5) *Apply edge tracking to finalize the detection by suppressing all the weak edges that are not connected to strong edges.*

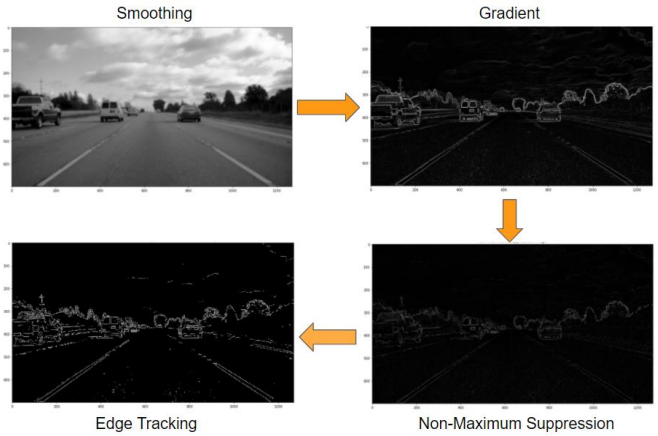

Figure 1 Canny Edge Detector

Figure 2 Hough Transform to Detect Lane

After successfully extracting edges from the original image, Hough Transform will be applied to detect straight lines in the binary edge map. Using the parametrization equation:

$$\rho = x \cdot cos\theta + y \cdot sin\theta$$

We could map each edge point in the image plane to a sine-like line in Hough Space. Then, the points in Hough Space that have the highest accumulated value would be considered as possible lines in the image plane. As shown in Figure 2, this method successfully detects the left and right lanes in an example image.

3.2 B-Spline and Canny/Hough Estimation of Vanishing Points

With the tools of the Canny edge detector and Hough transform implemented, the prerequisites for the B-spline method have been met. Before introducing the algorithm, it is important to state the assumptions of this approach:

1) *As stated in the introduction, this paper only focuses on detecting the lane centers of the lanes, which is based on the underlying assumption that the lane is always wider than the car.*
2) *There is no disappeared lane line, merging lanes and similar ambiguous situations.*
3) *Horizon is a horizontal line in the image.*
4) *Lane lines will never be horizontal in the image.*
5) *All vanishing points in the image belong to the same horizon.*
6) *The camera is fixed on the car, and the position of the horizon in every image is approximately the same and their approximate position is pre-known.*

With these assumptions, the lane's position can be represented by a B-spline drawn by the following steps:

1) *Conservatively chop off the part of the image above the horizon since the sky, tall buildings and trees are not relevant to lane detection. By assumption 6), this can be achieved easily.*
2) *Use a Canny edge detector on the input image, the output of this step will be used as an input for all the following steps.*
3) *Perform Hough transform to roughly identify lane lines. Since they are only an estimate, these lane lines will not be used as the final result. Instead, they will be used to create the horizontal strips in step 6).*
4) *From these lane lines remove the ones that are horizontal; they are outliers by assumption 4).*

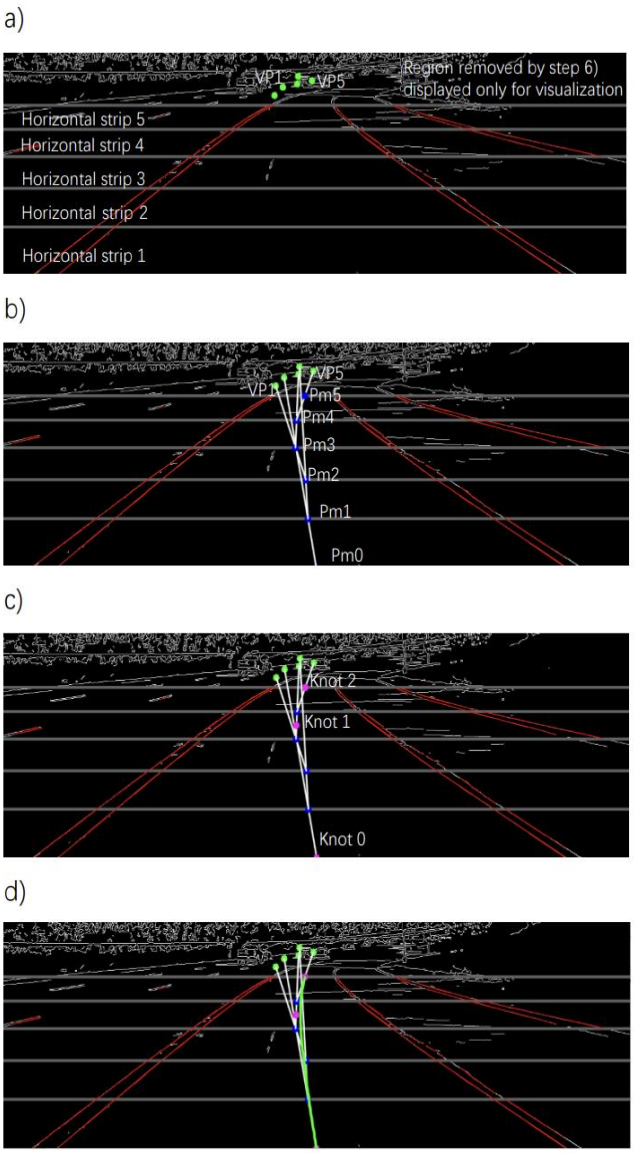

Figure 3 Intermediate steps for the algorithm
a) Five horizonal strips created by step 6) (separated by grey horizontal lines) and the vanishing points (VP1 to VP5) that correspond to the lane lines (in red) in each of the strip.
b) The control points in step 10), Pm0 to Pm5 (blue). For i-th control point Pmi is obtained by first drawing a line connecting Pm(i-1) and VPi, (shown by the white lines) and the i-th control point is the intersection of the white line and the upper bound of i-th horizontal strip.
c) B-spline knots (purple-pink)
d) Spline fitted (green curve)

5) *Obtain a better estimate of the location of the horizon by finding the most common vanishing point of the lane lines obtained by step 3) using RANSAC .*
6) *With a better estimation of the horizon, we can further remove the portion of the image above the horizon without being conservative. We also removed the part of the image near the horizon (even if it is below the horizon), as the lane lines near the horizon are too far from the camera and appear to be too small. As such, it turned out that Hough transform is more likely to pick up noise than these faraway lane lines. Note that the figure provided in this paper still includes this removed portion of the image, but it is only for clearer visualization; these removed portions will never be considered by the algorithm from now on. Then, segment the remaining portion of the image into five horizontal strips. Repeat step 7) and 8) for all five strips, resulting in five vanishing points on the same horizon corresponding to the lane lines in the five different horizontal strips. These vanishing points will be a good indicator of the direction of the lane lines in each of the strips, as they will be different in the case of a curved road. (Figure 3 a))*
7) *Perform Hough transform to identify lane lines in one of the horizontal stripes. As the height of each of the horizontal strips is much smaller than the height of the entire image, the lane lines can be approximated as being straight locally in each of the strips. Also, from these lane lines remove the ones that are horizontal; they are outliers by assumption 4). (Figure 3 a))*
8) *From the approximately straight lane lines in each of the strips, use RANSAC to fit a common intersection. This will be the vanishing point corresponding to the lane lines in the current strip. (Figure 3 a))*
9) *By assumptions 3) and 5) the heights of the vanishing points between strips will not differ too much, thus outliers can be rejected if the height of the vanishing point varies too much between strips.*
10) *Find the control points, should be six of them. The lowest control point is the center bottom of the image. Then for the other five control points, each control point corresponds to the five horizontal strips. Start by first considering the lowest strip and find the control points recursively. Draw a line segment connecting the previous control point and the vanishing point corresponding to the current strip. The intersection between the line segment and the upper boundary of the strip is the next control point. After finding the new control point, move on to the strip above the current strip. Mark these control points as Pm5 to Pm0. (Figure 3 b))*
11) *Find the knots for the B-spline, using the control points. Namely, for the control points shown in figure 3a), the knots are Pm5, the midpoint between Pm2 and Pm1 and Pm0. (Figure 3 a) and c))*
12) *Fit a B-spline through the knots (Figure 3 d))*

The above is an overview of the entire algorithm. What is original in this paper is beyond the fact that it is presenting an original implementation to an existing algorithm, there are many changes to the algorithm too, which includes:

1) *Assumption 1), 4), 5) and 6) are constraints that are original to this paper.*
2) *With assumption 4), the algorithm (step 4)) can further eliminate irrelevant lines generated by Hough transform before they have been passed in as inputs to RANSAC, making the algorithm more robust.*
3) *With assumption 6), irrelevant parts of the image can be easily removed, as contours of trees and clouds only give negative impacts to step 3).*
4) *[1] did not specify how to obtain the position of the horizon for step 6) so step 3) has been added to our algorithm.*
5) *Step 9) is original.*
6) *As this is a reimplementation of a closed source project, all the parameters used in this project are original. Namely, the five parameters that characterize the position of the strips, the two parameters for Canny and thirty parameters for Hough transform (the parameters for each time of Hough transform are different) three parameters for RANSAC and the five parameters used in step 4) and 9) combines to at least forty parameters used in this project. We have spent at least twenty-five hours tunning the parameters and making design tweaks listed previously in this section according to each of the disturbances we see. The model never worked while first implemented with random parameters. With these tunning and minor design changes, we achieved the results presented in the later section.*

4. Evaluations

To evaluate the performance of the Lane Detection Method, we choose 40 test images from [2]. As illustrated in Table 1, the images are sorted into different categories, and the same image can be marked with multiple labels. For example, some of the images contain curved lanes, while the others contain straight lanes. The lanes lines can also be dotted or solid. Also, to evaluate the robustness of our lane

detection algorithm against disturbance, we include occlusion and shadow as two additional classes.

Figure 4 Test Images

Figure 4 above is a visualization of part of the test images. We perform lane detection on these images and manually decide if our detector passes the test case and record the number of true positive cases in each class. Then, by dividing the true positive cases by the number of images in each class, we can calculate the success rates, as shown in Table 1.

	Total	Straight Lane	Curved Lane	With Occlusion	With Shadow	Solid Lane	Dashed Lane
Num of Test Case /Images	40	15	25	26	15	8	32
Success Rate	80%	86.67%	76%	76.92%	73.33%	100%	75%

Table 1 Testing Result

Since solid lanes can generally be detected perfectly by Canny/Hough Estimation in most of the cases, it achieves a success rate of 100% on test images, which is much higher than dashed lanes detection. However, only good at detecting solid lanes is not good enough since a large majority of the lanes on real roads are dashed lanes. To make our algorithm robust enough to a real-world situation, we need to improve the accuracy for dashed lane detection in the future. Moreover, curved lanes seem to be more difficult to detect than straight lanes. This is because perfectly straight lanes can be located much easier by Hough Transform. Additionally, occlusion and shadow can affect, though only to a limiting extent, the performance of our lane detection method by generating noisy edges. The noisy edges might further induce incorrect estimations to Hough Transform and eventually result in some weird vanishing points.

Although the performance of our lane detection algorithm is affected when the occlusion or shadow is presented, they still achieve a success rate of 76.92% and 73.33%, which is not bad considering there is a lot of disturbance in the scene. Figure 5 shows an example when there is some occlusion induced by other vehicles driving on the road. Figure 6 shows an example of when there are some shadows on the road. In both cases, our algorithm can successfully locate the lane directions. These two test cases illustrates the robustness of our lane detection method against disturbance.

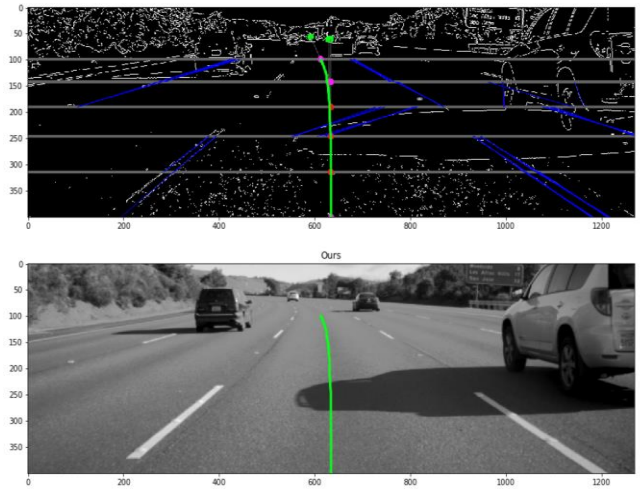

Figure 5 Detection Result (with occlusion)

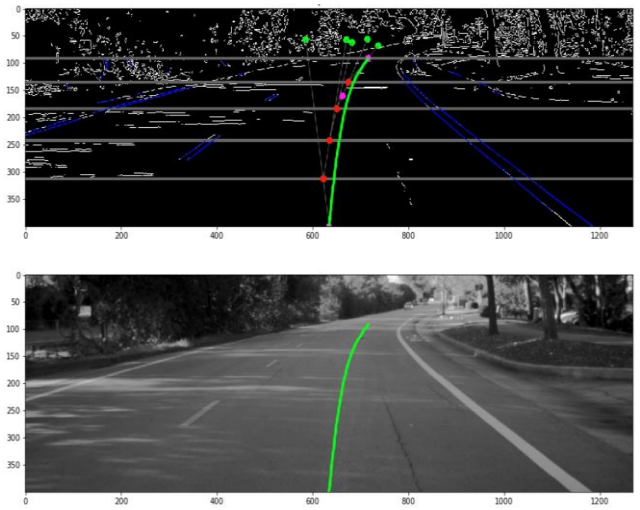

Figure 6 Detection Result (with shadow)

Conclusion

The lane detection algorithm proposed in this paper can accurately detect lanes under various scenarios (i.e., broken and solid lane lines, curving and straight roads). It shows robustness to disturbances in the scene, especially when the occlusion and shadow are presented. Also, the method does not rely on a vast number of labeled datasets. Therefore, it

has the advantage over the other methods which involve a large amount of computational expensive training.

One of the potential improvements to the proposed algorithm is to tackle the issue that the proposed method did not take any advantage of the colors in the image. The input to the software is a greyscale image of the road, which already lost certain information about the lanes. Future iterations of the method can take advantage of the color information to enhance robustness.

In addition, tuning better parameters for the model is always a potential improvement. From the progress we have been constantly achieved through parameter tuning, we are expecting to halve the failure rate with another thirty hours of parameter tuning.

Meanwhile, [1] did not use Canny/Hough Estimation of Vanishing Points alone to calculate the knots for a B-spline. Instead, the method was used in combination with minimum mean square error. Implementing an additional method to have two methods supervise each other would be a potential improvement to robustness.

Lastly, the input used in this paper is images of lanes, but the problem can be potentially easier if the inputs are videos. This is because the lane directions do not change much between frames, so the result obtained for each of the frames can regulate each other to reject outliers.

Code

https://github.com/shunyaoxu/Stanford_CS231A_Project_Lane_Detection.git

References

[1] Yue Wang, Eam Khwang Teoh, Dinggang Shen, Lane detection and tracking using B-Snake, Image and Vision Computing,Volume 22, Issue 4,2004,Pages 269-280,ISSN 0262-8856

[2] Behrendt, Karsten and Soussan, Ryan, "Unsupervised Labeled Lane Markers Using Maps", llamas2019, Proceedings of the IEEE International Conference on Computer Vision, 2019.

[3] Kawache, 2021)“Python-B-spline-examples” https://github.com/kawache/Python-B-spline-examples#readme

[4] OpenCV, “Hough Line Transform”,Mar,2022, https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html