

VANISHING POINT AIDED LANE DETECTION USING A MULTI-SENSOR SYSTEM

Zifan Zhang¹, Gyoungmin Kang¹, Mengchi Ai¹, Naser El-Sheimy^{1*}

¹Dept. of Geomatics Engineering, University of Calgary, 2500 University Dr NW, Calgary, AB T2N 1N4, Canada –
(zifan.zhang1, gyoungmin.kang, mengchi.ai, elsheimy)@ucalgary.ca

KEY WORDS: Lane Detection, Image Processing, Sensor Fusion, LIDAR, Multi-sensor System

ABSTRACT:

Lane Detection is a critical component of an autonomous driving system that can be integrated alongside with High-definition (HD) map to improve accuracy and reliability of the system. Typically, lane detection is achieved using computer vision algorithms such as edge detection and Hough transform, deep learning-based algorithms, or motion-based algorithms to detect and track the lanes on the road. However, these approaches can contain incorrectly detected line segments with outliers. To address these issues, we proposed a vanishing point aided lane detection method that utilizes both camera and LiDAR sensors, and then employs a RANSAC-based post-processing method to remove potential outliers to improve the accuracy of the detected lanes. We evaluated this method on four datasets provided from the KITTI Benchmark Suite and achieved a total precision of 87%.

1. INTRODUCTION

1.1 Background

With more robust algorithms and advanced sensor technologies, autonomous vehicles (AV) are becoming more accessible and adept to deal with complex environments. One of the emerged concepts to improve and aid autonomous driving is High-definition (HD) map, which provides high precision and informative level of geospatial information. A HD map contains critical properties of the road network such as roads, traffic lights, markings, and signs. Lane-level maps are especially significant for the construction of HD map as it provides semantic segmentation of the road roads and lanes (Zhou et al., 2021). Such crucial steps can be achieved through lane detection techniques as it enables the localization of the road and establishes the relative position of the autonomous vehicle.

Many researches have explored the approach of lane detections using sensor technologies. (Low et al., 2014) proposed an optimized Canny edge detection and Hough Transform to detect the left and right lane markers using front-viewed camera system. (Kim et al., 2018) used another method with stereo camera alongside with Dynamic programming and the Kalman filter to trace the white line markings to detect lanes. Some attempted to use LiDAR sensor to directly detect lane features in 3D environment. (Huang et al., 2021) proposed a constrained RANSAC algorithm to select the road region, then apply a road curb detection method and an adaptive threshold selection to identify the lane boundaries from LiDAR point cloud.

Some also attempted both or more sensors, (Bai et al., 2018) used a multi-sensor system to extract autonomous navigation features such as road curb detection, lane detection and traffic sign recognition using vision-based lane detection, Z-variance method and Haar-like feature-based method. (Zou et al., 2020) evaluated deep-learning-based algorithm that achieved lane detection in typical scenarios. However, there are still challenging problems with these techniques where the detected lane and road features can contain outliers that are hard to eliminate, and difficult to fuse with other sensor frameworks in an autonomous driving system.

To address these issues, we propose a method that can be summarized in two configurations. First, we use vanishing point aided method as constraints to detect boundary and inner lanes and apply coordinates transformation to standardize the information. Then, we use a RANSAC-based algorithm to estimate the best parametric line that closely resembles straight boundary lanes to remove potential outliers. The proposed solution is evaluated based on KITTI datasets (Geiger et al., 2013), and the results obtained are evaluated with manually labelled data.

2. RELATED WORK

Lane detection is a crucial part of autonomous driving systems that has received significant attention from researchers. To ensure road safety and improve robustness of lane detection performance, numerous studies have been conducted. These studies have led to development of reliable and effective techniques, ranging from traditional computer vision methods to deep learning-based methods.

Traditional computer vision techniques have been widely used to detect lane markings on the road surface, such as Canny edge detection and Hough transform (Low et al., 2014). While our proposed method also employs these techniques, simply averaging the detected lines is not a reliable solution. This approach can lead to noisy environments and can only detect two dominant left and right lanes. Additionally, averaging the lines can lead to significant errors if a false line is identified, which can cause the lane location to be skewed.

Another similar approach leverages the vanishing point in addition to Canny and Hough transform was attempted in (Youjin et al., 2018) and (Kong et al., 2009). The methods involve estimating the vanishing point where parallel lane lines appear to converge at the end of the road in the image. To detect the vanishing point, (Youjin et al., 2018) computed the intersection of all the lines detected to find the most common intersection point, which requires significant processing time, particularly in noisy environments. To reduce the processing time, we propose a new approach that first identifies the two outer road boundary

* Corresponding author

lanes using lane properties first and then simply intersects them to locate the vanishing point.

In terms of deep learning-based algorithms, previous works have shown promising results in lane detection. For instance, (Zou et al., 2020) developed their algorithm by integrating the convolutional and recurrent neural networks (CNN and RNN). CNN is responsible for extracting lane features from a single frame, which are subsequently sent to RNN for lane prediction. This approach is robust and provides accurate results, however, it can be time-consuming and its result cannot be used to generating a HD map.

Furthermore, approach that utilizes both camera and LiDAR sensors has also been proposed for lane detection. This method involves implementing a deep neural network to directly estimate lane markings as well as the ground height in 3D space (Bai et al., 2018). While this approach is efficient and provides accurate results in many cases, it struggles when there are heavy occlusions, or some frames are missing lane marks. To overcome this issue, we propose using k-mean algorithms and RANAC to remove outliers and construct continuous lanes.

(Li et al., 2014) proposed a feature-level fusion method from the LiDAR and vision data to detect the lanes. The method first using fusion data to detect curb points, then using two more layers to filter the data and define the optimal driveable area. Then, the driveable area's image is processed using vision-based algorithms to detect lanes. However, the optimal driveable region is limited due to other obstacles along the road.

To address these challenges, we proposed a different approach to detect lane markings rapidly by rigorous pre-processing leveraging the vanishing point, followed by post processing using a multi sensor system to accurately detect lanes that can be used for HD map.

3. METHOD

3.1 Overview

Figure 1 illustrates the workflow of the processing steps, we first attempt to identify the two outer road boundary lanes using images captured by cameras and image processing techniques with noise reduction threshold. The vanishing point is identified as constraint, which would be the convergence point of lane lines in the context of lane detection to detect any inner lanes through a mask region created. The vanishing point was identified simply by intersecting the two boundary lanes. The LiDAR point cloud is processed using RANSAC to extract a parametric plane that resemble the road surface, which is used for 3D-reconstruction and coordinate transformation of the output into local ENU frame and global frame. Finally, the post-processing is applied on the detected lanes and road features in the local ENU frame.

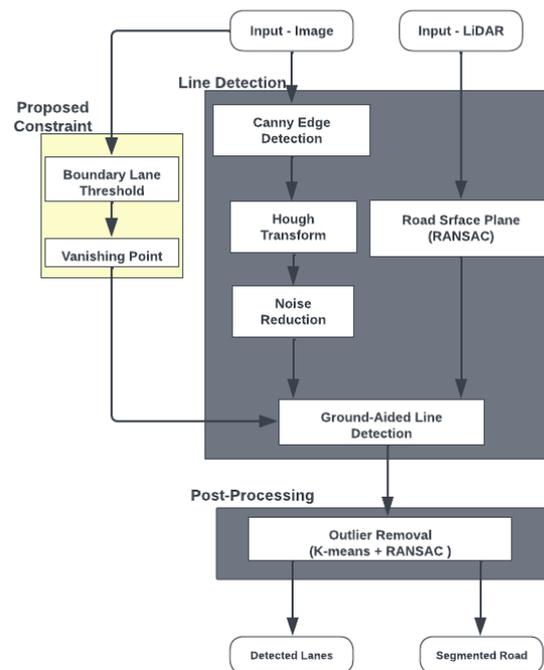


Figure 1. Flowchart of the proposed method implementing vanishing point based constraint and post-processing step

3.2 Coordinate System Definition

This section outlines the local and global coordinate frames used by transformation, post-processing, and trajectory odometry visualization. Each sensor's local frames are defined in the KITTI Benchmark Suit's setup.

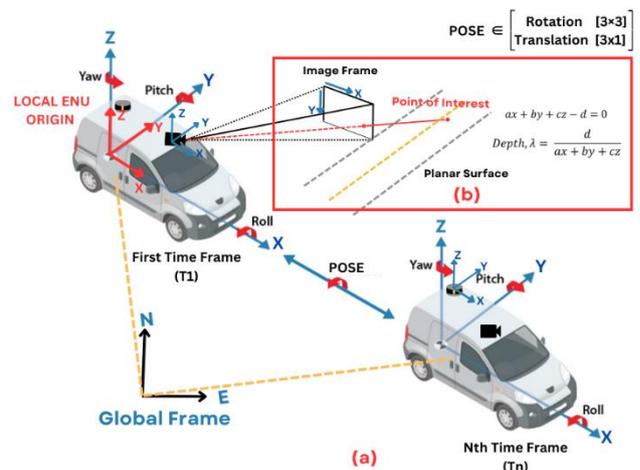


Figure 2. Coordinate system definition of the setup with the image, local sensor, ENU and global frames.

Figure 2 illustrates the camera, LiDAR sensor and INS/body frame definition, as well as the orientation of local ENU frame and the global frame. Prior to data collection used for the KITTI Benchmark Suite, all sensors have been calibrated, and the calibration parameters for each specific dataset were provided, including the projection matrix, extrinsic and intrinsic parameters for the digital camera, as well as the rotation and translation

information between each sensor within the established rigid body system.

Local ENU

To perform post-processing, the coordinate frame is first transformed into the body frame, with the origin aligned on the INS sensor. Then, the orientation and position information obtained from the INS are used to obtain the pose information to transform points into a local ENU frame. The origin of this frame is the body frame of the first frame of the evaluated dataset.

(1) is used to transform a point x in the body frame to the local ENU frame. (2) is the transformation matrix's construction. This matrix is used to transform the coordinates from the body frame to the local ENU frame.

$$X_{ENU} = T_{Body}^{Local\ ENU} X_{local} \quad (1)$$

$$T_{Body}^{Local\ ENU} = [R; t; 0\ 0\ 0\ 1] \quad (2)$$

where $T_{Body}^{Local\ ENU}$ = transformation matrix
 X_{local} = coordinates in local sensor frame
 X_{ENU} = local ENU frame origin at the first frame's body frame
 R = rotation matrix
 t = translation vector

(3) and (4) compute the rotation and translation information using the INS readings. $X_{Ground\ Truth}$ and $Y_{Groun\ Truth}$ are first transformed from latitude and longitude.

$$R = R_x R_y R_z \quad (3)$$

$$t = \begin{matrix} X_{Ground\ Truth} \\ Y_{Groun\ Truth} \\ h \end{matrix} \quad (4)$$

where $R_x R_y R_z$ = rotation matrices based on roll, pitch, and yaw
 $X_{Ground\ Truth}, Y_{Groun\ Truth}$ = coordinates in Mercator coordinates frame
 h = altitude

Global Coordinate System

The local ENU frame obtained can be transformed into ground truth frame. To obtain the positioning information, the INS data is measured based on WGS84 datum. By following the approach from (1), the latitude and longitude information can be derived from the local ENU coordinate frame. To align with the ground truth frame, the inverse of the transformation matrix from (2) is used.

3.3 Vanishing Point Aided Lane Detection

As illustrated in Figure 1, lane detection begins with applying Canny edge detection, followed by Hough transform. Canny edge detection mainly involves identifying sudden changes in intensity values between neighbouring pixels to detect edges. When implementing Canny edge detection (Li et al., 2020), it was critical to choose the proper sigma, which is known as the standard deviation of the Gaussian filter; a smaller filter detects small and sharp lines, whereas a large filter applies more blurring,

thus it needs longer processing time. Another important parameter needed to adjust was "aperture size" when calling the Canny function. It indicates the order of Sobel filter used to compute the gradient. In order to capture more detailed edge information, the aperture size was specified as 5 so that possible lane edges cannot be missed. Once all edges were detected, Hough transform was applied to detect straight lines among the edges, which is an effective feature extraction technique to find a certain shape such as a circle or line by voting program (Li et al., 2020). The line features extracted by Hough transform, however, contained a lot of noise as shown in Figure 3.

Thresholds: To remove all non-lane markings, four thresholds were established based on the characteristic of lanes:

- **Line Direction:**
When considering the two road boundary lanes, they cannot be either vertical or horizontal, but instead lie at an angle between 15° and 75° from the x-axis (Fig. #).
- **Y values:**
As lanes always exist in the bottom part of an image, the y pixel value was used as constraints. Any lines having the y value below one third of the image height were discarded, while lines having the y value close to the image height were kept.
- **Minimum line length:**
The minimum line length to keep was set to 40 pixels.

$$l = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (5)$$

$$l > 40$$

where l = detected line's length
 x_1, y_1, x_2, y_2 = image coordinates

- **Location:**
Since lanes can be divided into left and right lanes, each line can be determined as either a left or right lane based on its slope. If a line has a negative slope, but is located on the right side of the image, it will most likely be a false lane and vice versa for a line with a positive slope.

$$s = \frac{y_2 - y_1}{x_2 - x_1} \quad (6)$$

where s = slope
 x_1, y_1, x_2, y_2 = image coordinates

$$\frac{Image\ width}{2} + 50(buffer) > x_1\ and\ slope < 0 \rightarrow left\ lane \quad (7)$$

$$\frac{Image\ width}{2} + 50(buffer) < x_4\ and\ slope > 0 \rightarrow right\ lane \quad (8)$$

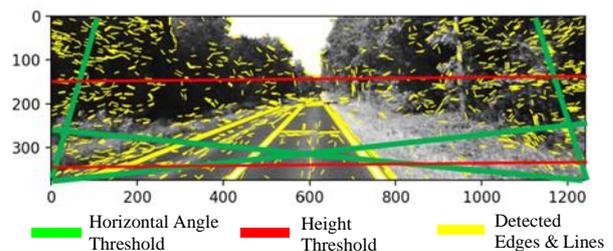


Figure 3. Detected edges and lines on the image before noise reduction and vanishing point.

After filtering out lines that did not satisfy all four threshold values, we chose the two road boundary lanes based on both their slope angle and y-values. The filtered boundary lines were stored as the left and right lanes and determined the boundary lane for each side as the line with the minimum y-value of the end point and the start point, respectively. However, to avoid mistakenly identifying the middle lane as an outer lane, we used an angle criterion based on the line with the smallest horizontal angle and a small y-value. If the difference between the minimum value and the line's y-value was less than 10 pixels, it was identified as the outer lane.

Vanishing point: The vanishing point was detected by intersecting the two outer lanes, and then the middle lanes were identified by finding lines passing through it. Non-lane markings that pass through the vanishing point were eliminated using a lane width threshold, requiring the x-intercept of the outer lanes and middle lanes to be larger than 350 pixels.

3.4 3D-Reconstruction

The detected lanes in the image frame are reconstructed into a 3D frame by utilizing the road surface as a reference plane to estimate the depth information. The lanes are assumed to be located on the road surface, which leads to the acquisition of similar height information. Fig. 4 depicts the LiDAR points that are initially classified as ground points by considering the vehicle's dimension. As the segmented ground points are assumed to follow a parametric plane model that closely resemble a road surface, the RANSAC algorithm is employed to determine the best-fit plane that estimates the road surface at the evaluated time frame.

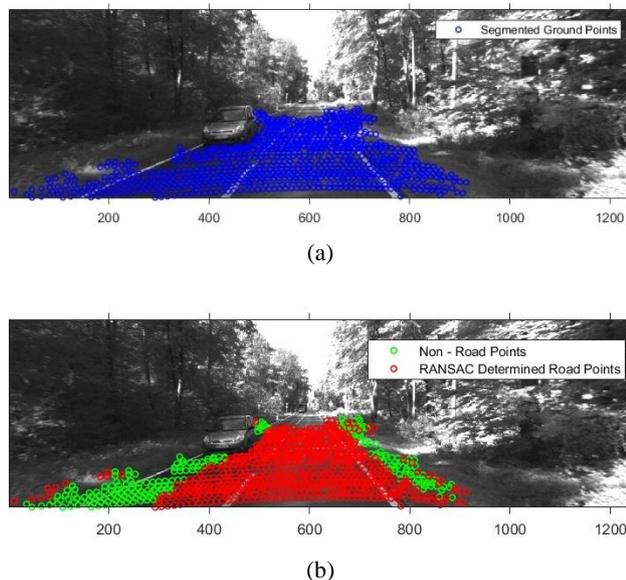


Figure 4. Segmented ground points and road surface plane

Once the model plane parameters are obtained, they are used to estimate the depth information using planar holography technique. Figure 2 (b), shows any points that are assumed to be on the road surface can be re-projected onto the road surface plane by computing the scale parameter, λ .

3.5 RANSAC-Based Outlier Removal

The final fused detected lanes in the evaluated dataset comprise the two outer boundary lanes and inner lanes. When transformed into the local ENU frame, the lanes are not grouped or indexed, and they are subject to outliers and noise. It should be noted that the proposed method is tested mainly in rural areas where the road is mostly straight with minimal curvature and in a limited time frame. Thus, the effectiveness of the proposed method may not work well in urban environments with more intricate environment.

Performing the post-processing in the local ENU frame can be challenging as outliers may obscure the data sequence. To address this, we apply the K-means clustering algorithm (Forgy, 1965) to group detected boundary lane points into two clusters, representing the left and right boundaries of the lane. Each set of grouped lane points is then evaluated using a modified RANSAC algorithm to determine a 3D best-fit line that passes through the identified inliers, resulting in a polished detected straight lane.

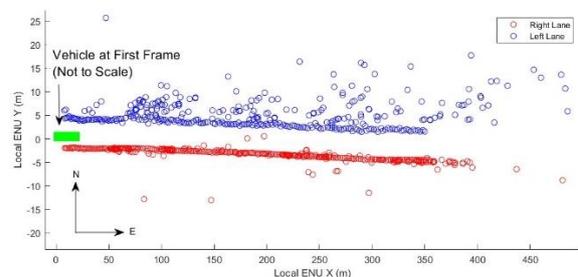


Figure 5. Grouped left and right lanes evaluated in the local ENU frame.

The remaining inliers are re-fitted with a 3D line model to construct a best-fit line that resemble a single lane. The experimental result of this process is presented in the next section.

4. EXPERIMENTAL RESULTS

We evaluated our lane detection tool using The KITTI Vision Benchmark Suite, which provides a comprehensive dataset including images, LiDAR, and INS data. Initially, we used the "2011_09_26_drive_0027" dataset from the raw road category, which was collected on a simple two-lane straight road in a rural area.

Dataset	Number of frames	Description	TP	FP	FN	Accuracy(%)	Precision(%)
2011_09_26_drive_0002	76	Urban areas with train tracks and bike lane	139	32	65	59	81
2011_09_26_drive_0015	296	Countryside with train tracks and some incoming cars	464	66	203	63	88
2011_09_26_drive_0027	187	Straight road with less incoming cars	456	11	108	79	98
2011_09_26_drive_0028	429	Curvy road with many incoming cars	201	72	375	31	74
Total			1260	181	751	57	87

Table 1. KITTI dataset evaluation result with data descriptions

Once the algorithm provided accurate results with this dataset, we applied it to more datasets for testing, which provided more diverse and intricate environments. The algorithm was then further refined to ensure robustness and accuracy with the new data set.

4.1 Results

The images in Figure 7 shows the results of applying the vanishing point constraint and noise reduction threshold requirements to the training dataset. The boundary and inner lanes were determined separately and visualized in cyan and blue, respectively. While inner lanes are typically drawn with a dotted line, we presented them as a straight continuous line to improve visualization by connecting them all the way to the vanishing point, shown as a green dot.

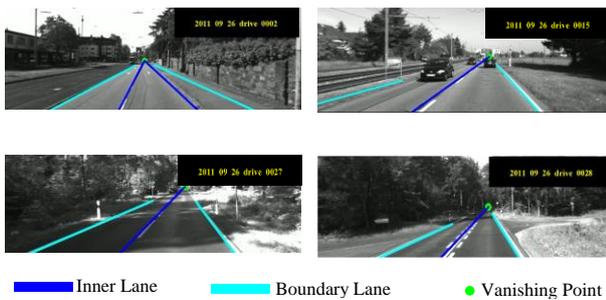


Figure 6. Detected boundary and inner lanes

Figure 7 shows the detected lanes and the corresponding road features in both image and LiDAR frames. The detected lanes are represented by green lines, and the corresponding road feature is labelled in red.

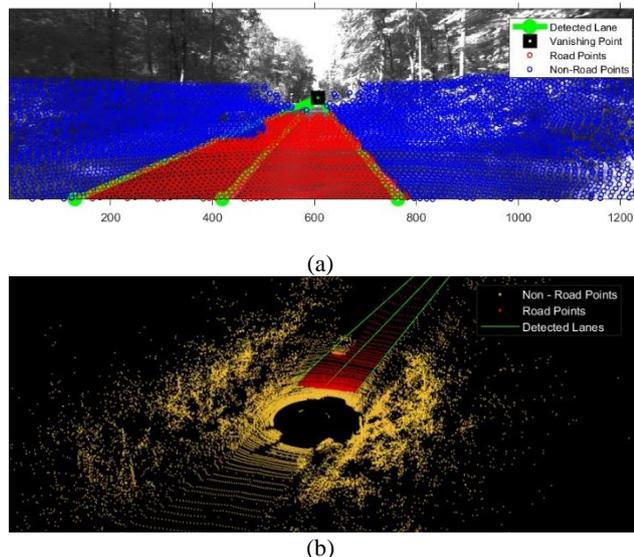


Figure 7. Detected Lanes and extracted road features represented in image frame (a) and LiDAR frame (b). Note that in (b), the vanishing point is not visible in the captured image due to zoomed-in affect

Fig. 8, shows the final output of the evaluated dataset, “2011_09_26_drive_0027”, including the final detected boundary lanes projected onto the ground truth frame, local ENU frame and the labelled road features based on the detected lanes from the overall LiDAR point clouds.

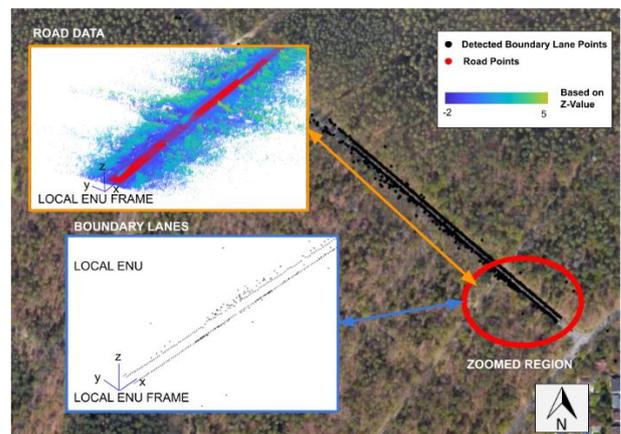


Figure 8. Detected lanes and road transformed in the local ENU frame and global frame

Figure 9 shows the post-processing results obtained in birds-view, the blue points represent the outliers, green points represent the inlier, and the red lines are the final corrected lanes.

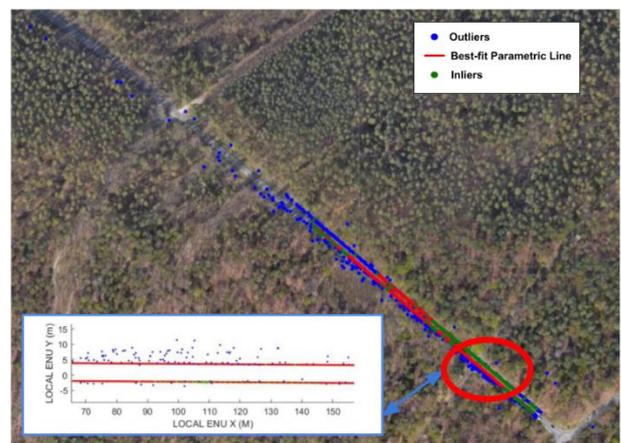


Figure 9. Post-processing results in (a) local ENU frame (top-down view) and (b) global frame

4.2 Comparison

CV-Based Algorithm

One common state-of-art approach is to use simple Canny edge detection alongside Hough Transform to detect any lanes based on images. Using this algorithm on KITTI dataset of 0027, frame 04, as an example, the region of interest is first selected and masked out to perform canny edge detection and Hough Transform. The selection mechanism is simply constructed by creating a triangle shaped area from the centre to the bottom two corners of the image.

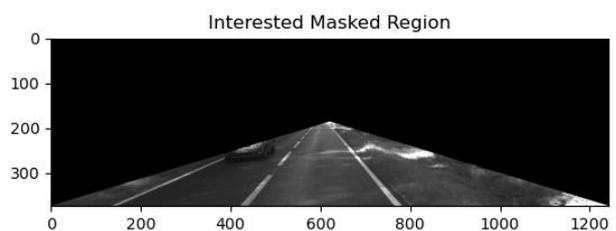


Figure 10. Assumed interested region

The results are prominent in that it contains many noises, errors, and mistakes around the grass area near the road.

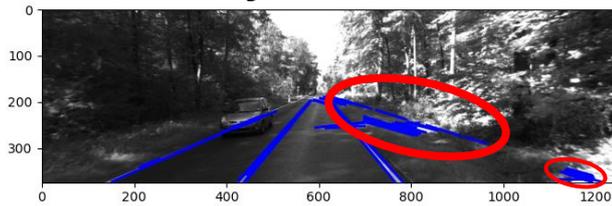


Figure 11. Detected edges and lines with noises and significant outliers.

The limitation of such a simple and efficient algorithm is that it cannot handle complex environments, especially when there are other environmental factors such as grass in the case of “2011_09_26_drive_0027” dataset. It was originally designed for highway environment when most of the road environments consist of road features.

4.3 Evaluation

To evaluate the performance of our lane detection algorithm, we employed the MATLAB Ground Truth Labeller App. This evaluation approach involved a semi-automated process, where the true lanes were manually labelled by human vision to compare against the detected lanes in a particular dataset.

The evaluation of detected lanes composes confusion matrices based on comparing the predicted lane markings with the ground truth markings identified by visual inspection. To determine whether a predicted lane was accurate, we compared the slope and y-intercept of the detected line against those of the ground truth marking, using threshold values of 0.5 for the difference in slope and 100 pixels for the difference in y-intercept. True Positive (TP), the number of correctly detected lane markings. False Negative (FN), the number of missed lane markings, and False Positive (FP), the number of falsely detected lane markings. By analysing these metrics, the accuracy and precision can be calculated using the following equations:

$$accuracy = \frac{TP}{TP+FP+FN} \quad (9)$$

$$precision = \frac{TP}{TP+FP} \quad (10)$$

As can be seen in Table 1, it is evident that lane detection algorithm showed the highest performance on the “2011_09_26_drive_0027” dataset, with an accuracy of 79% and precision of 98%. This dataset was mainly used during the development of the algorithm, which could be a contributing factor to the superior result.

However, the accuracy dropped significantly to 31% on the “2011_09_26_drive_0028”, which contained curvy roads and heavy traffic, making it challenging to detect lanes correctly. On the other two dataset, the algorithm proved its robustness by identifying lanes accurately, even in the presence of train tracks in the images which can lead to more false positives. Consequently, the overall accuracy was calculated to be 57%.

5. CONCLUSIONS

In this paper, we proposed a vanishing point aided lane detection method, using camera and LiDAR sensors, and a RANSAC

based post-processing method to remove outliers. By leveraging the characteristics of lanes, we successfully identified the lanes and vanishing point in noisy environments. Additionally, implementing RANSAC and k-means clustering algorithms effectively removed outliers among the detected lanes. Although the accuracy of the lane detection algorithm in image processing was 57%, we expect that the accuracy of the final parametric line will increase significantly after the outlier removal process.

However, there are some limitations that need to be addressed in future work. One of the limitations is its inability to detect curved inner lanes as they would no longer pass through the vanishing point. One of the assumptions we made is that roads are straight so inner lanes will always pass the vanishing points. To overcome this limitation, we suggest considering the location of the vanishing point to identify winding roads and apply a different method to detect the curvy inner lanes. The post-processing assumed the final detected lanes are straight, which also imposes limitations on its functionality when handling curvy roads.

For future work, the focus can be shifted on accommodating curvy lanes. To identify the curvy lanes, the direction of the curve should be considered based on the location of the vanishing point. This can be achieved by performing curve line fitting using the vanishing point and inner lines detected. Furthermore, the system can use other relevant information as constraints such as curbs to further improve lane detection accuracy. The post-processing method can be optimized to adapt non-straight lines using more complexed parametric model.

ACKNOWLEDGEMENTS

We would like to extend our appreciation to Bo Xie and Farid Hode for their assistance in completing this project. Additionally, we would like to acknowledge the KITTI Vision Benchmark Suite for providing the datasets used in this research.

REFERENCES

- Bai, M., Mattyus, G., Homayounfar, N., Wang, S., Lakshmikanth, S.K., Urtasun, R., 2018. Deep Multi-Sensor Lane Detection, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3102–3109.
- Forgy, E., 1965. Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Biometrics*.
- Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research* 32, 1231–1237.
- Huang, J., Choudhury, P.K., Yin, S., Zhu, L., 2021. Real-Time Road Curb and Lane Detection for Autonomous Driving Using LiDAR Point Clouds. *IEEE Access* 9, 144940–144951.
- Kim, J.-G., Yoo, J.-H., Koo, J.-C., 2018. Road and Lane Detection Using Stereo Camera, in: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp). Presented at the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 649–652.
- Kong, H., Audibert, J.-Y., Ponce, J., 2009. Vanishing point detection for road detection, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2009

IEEE Conference on Computer Vision and Pattern Recognition, pp. 96–103.

Li, Q., Chen, L., Li, M., Shaw, S.-L., Nüchter, A., 2014. A Sensor-Fusion Drivable-Region and Lane-Detection System for Autonomous Vehicle Navigation in Challenging Road Scenarios. *IEEE Transactions on Vehicular Technology* 63, 540–555.

Li, X., Yin, P., Zhi, Y., Duan, C., 2020. Vertical Lane Line Detection Technology Based on Hough Transform. *IOP Conference Series: Earth and Environmental Science* 440, 032126.

Low, C.Y., Zamzuri, H., Mazlan, S.A., 2014. Simple robust road lane detection algorithm, in: 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS). Presented at the 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS), pp. 1–4.

Raj, A., Krishna, D., Hari Priya, R., Shantanu, K., Niranjani Devi, S., 2012. Vision based road surface detection for automotive systems, in: 2012 International Conference on Applied Electronics. Presented at the 2012 International Conference on Applied Electronics, pp. 223–228.

Xu, Z., Baojie, X., Guoxin, W., 2017. Canny edge detection based on Open CV, in: 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI). Presented at the 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), pp. 53–56.

Youjin, T., Wei, C., Xingguang, L., Lei, C., 2018. A Robust Lane Detection Method Based on Vanishing Point Estimation. *Procedia Computer Science, Recent Advancement in Information and Communication Technology*: 131, 354–360.

Zhou, Y., Takeda, Y., Tomizuka, M., Zhan, W., 2021. Automatic Construction of Lane-level HD Maps for Urban Scenes.

Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., Wang, Q., 2020. Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks. *IEEE Transactions on Vehicular Technology* 69, 41–54.