2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

# An Accurate and Computational Efficient System for Detecting and Classifying Ego and Sides Lanes Using LiDAR

Lucas de Paula Veronese, Asad Ismail, Vikram Narayan and Mathias Schulze

*Abstract*— **In this work, we are proposing a computationally efficient LiDAR based lane detection system that detects both ego and side lanes using 3D LiDARs. Our solution relies on the construction of local gird map around the ego vehicle using the infrared reflectance of combination of LiDARs. To fuse the information of the LiDARs into a map, the vehicle ego-motion is taken into account. The system is built using image processing by binarizing the map to extract the lane markers. The evaluation of computational performance of the final solution is realized on a single ARM core of the NVIDIA Drive PX2 without the need for the GPUs, and achieved a frame rate of 40 Hz. In the absence of a publicly available annotated dataset for LiDAR based lane detection, we evaluate the proposed solution against our proprietary camera based lane detection system. We observed a good correlation between the two in terms of Jaccard and Dice Coefficients.**

## I. Introduction

Several recent advances in affordable and effective perception technologies have given a major thrust towards the realization of autonomous vehicles. In this regard the intelligent vehicles research community have proposed and worked on several robotic systems which could eventually replace the human driver. To achieve this goal, the Defense Advanced Research Projects Agency (DARPA) established the Grand Challenge in 2004; a research program designed to trigger the advancement of technologies required to create a fully autonomous vehicle [1, 2]. Later several original equipment manufacturers (OEMs) in collaboration with Tier-1 suppliers have put in place advanced driver assistance systems (ADAS) like: adaptive cruise control, lane keeping, automatic emergency braking etc. into production. They are still called assisted (and not autonomous) functions, because under some adverse circumstances human intervention might be necessary.

There are examples where OEMs and academic research groups have achieved some degree of vehicle autonomy under ideal driving conditions [3, 4, 5, 6, 7]. Such research vehicles which are able to demonstrate higher level of autonomy are generally equipped with the most expressive sensor and high power computing units. The technical goal therefore is to replicate the same level of performance of the ADAS functions using a combination of less expressive (and less expensive) sensors and a low power computing device. In this case, the camera-based lane detection function can be considered as a successful example.

There are several commercially viable front facing monocular camera based lane detection systems, which now have a high degree of reliability. Most of the successful and contemporary camera based lane detection approaches are based on deep learning architectures [8, 9]. These approaches rely on a large amount of annotated dataset to learn the classifier that can further generalize on unseen data. Despite its effectiveness, the deep learning approaches sometimes require intense computational resources.

In order to achieve the goal of autonomous vehicles, we are required to increase the robustness of the functions by not relying on a single sensor input for decisions. Hence, realizing functions like Lane Detection using LiDARs acquire high importance [10, 11]. The contemporary LiDARs are more expensive than automotive grade cameras, but it is forecasted that with improvement in optical technologies the price of the LiDARs will come down drastically. Cameras can inherently perceive the color, while LiDARs are mainly designed to perceive the depth or distance in the 3D space. Thus, this aspect helps to make these sensors complementary to each other. Additionally LiDARs are more inured to lighting and illumination conditions, thus making it more robust compared to cameras. An alternative in this regard is to consider HD Maps as an intelligent sensor, which have detailed lane and road furniture information. However, the high cost involved in creating the map and availability of the map for all the geographical locations acts as a deterrent to substitute camera or LiDAR sensors by HD Maps. In this work, we propose a LiDAR based lane detection system that detects both ego and side lanes using 3D LiDARs. Refer to Fig. 1 for the detailed architecture of our LiDAR based lane detection.

The solution encompasses the detection of ground points and further clustering and classifying them as lane candidate or otherwise. This comprises the usage of image morphological operators for clustering of projected ground points and the vehicle ego motion information to achieve the fusion of detected lane candidate points over consecutive frames. The proposed approach is model based, and hence does not require a huge set of annotated training data. The final solution is realized on a single ARM core of the NVIDIA Drive PX2 without the need for GPUs, and achieved a frame rate of 40 Hz. To the best of our knowledge, this is the first LiDAR based Lane Detection approach which detects both the ego lanes and the side lanes, while most of the existing approaches [10, 11] detect only the ego lane. The experiments involved VLP 16 LiDARs, and resulting lane detection was compared against an in house camera based lane detection solution. We could observe that the proposed approach has a high degree of statistical

Lucas de Paula Veronese, Asad Ismail, Vikram Narayan and Mathias Schulze are with Visteon Electronics Germany GmbH, An der RaumFabrik, 33b, 76227 Karlsruhe Germany (corresponding author phone: +49 721 4766-1099; e-mail: lverones@visteon.com).

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

correlation with our proprietary camera based lane detection solution.

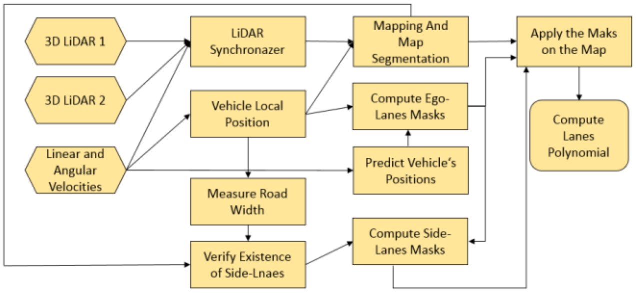


Fig. 1: The proposed LiDAR Lane Detection Architecture. Inputs from multiple LiDARs are synchronized and used for mapping and segmenting the ground points to detect possible lane candidates. Linear and angular velocities are used to calculate and predict the pose of the vehicle which in turn is used to calculate morphological masks used for detecting the ego lanes and side lanes. These masks are applied on segmented ground points and a regression function is subsequently used to fit a quadratic polynomial curve.

## II. PROPOSED LiDAR LANE DETECTION

The lane detection and classification proposed in this work uses a combination of two 3D-LiDARs. However, this is a scalable solution and can work even with a single or multiple 3D-LiDAR inputs. The proposed framework can receive one or multiple LiDARs inputs and further synchronize them in consonance with the linear ($v$) and the angular ($\omega$) velocities of the car. The vehicle odometry information is also integrated to create a local map within a short temporal frame. This grid map is converted into an image and is then binarized using image thresholding to emphasize the lanes. To identify the lines of each lane from image, our assumption is that the car is traveling mostly in middle of a lane. Therefore, masks for detecting lanes are built by extending from both sides of car up to 1.7 meters. The masks also covers the car pose $X = (x, y, \theta)$, from $t$ - 15 previous positions to $t + 15$ predicted ones. For the previous poses, the time difference $\Delta t$ is given by the subtraction of two consecutive time-stamps of the Velodyne's frames. The predicted pose of car is calculated using car motion model given by Eq. (1).

$$\begin{aligned} x_t &= x_{t-1} + v\Delta t\cos(\theta_{t-1}) \\ y_t &= y_{t-1} + v\Delta t\sin(\theta_{t-1}) \\ \theta_t &= \theta_{t-1} + \omega\Delta t \end{aligned} \qquad (1)$$

An important feature of our method is, since we use previous and predicted pose of the car if the car is moving in a bent or curved path the constructed masks are also curve, so we are able to find curve lanes in addition to straight lanes. Fig. 2 presents different masks on straight road, heavily and slightly bent, and 90° curve. To detect the existence of side lanes, the road width is calculated by extending a ray from the car side until it hits a white pixel on the image. If the road size is bigger than the size of one lane, the masks are applied on the side lane of the image. In the end, for every line detected by the previous steps of algorithm, the regression is called to compute the polynomial coefficients.

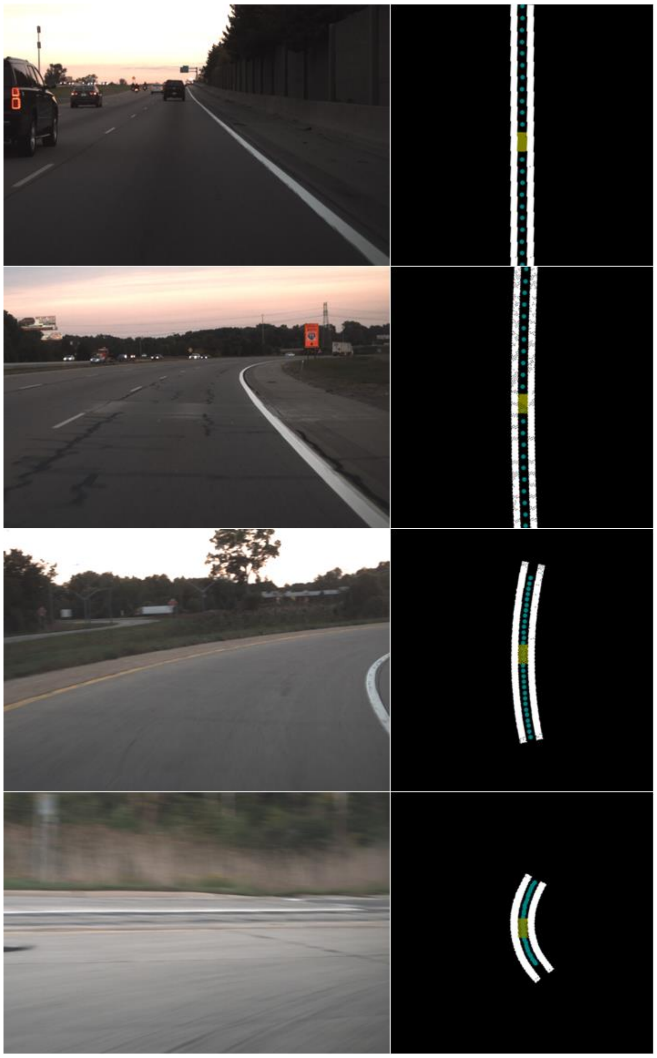


Fig. 2: Sequence of masks built to detect right and left boundaries of the ego-lanes. On the left is the camera image, and the corresponding LiDAR morphological mask is presented on the right. The camera image is used only for the sake of visual illustration of the road curvatures, and thus highlights the adaptive nature of our mask computing technique. The white lines show the masks for a straight road, a slightly bent, a heavier bent and 90° curve. The yellow rectangles represents the current positions of the ego vehicle on each mask The blue dots below and above the rectangles, (i.e., car pose), illustrate the 15 previous positions of the car and the 15 predicted positions of the car used to build the mask. The distance between the car pose and mask size are different for each mask shown since the speed is higher or smaller in some circumstances.

### A. Ego-Lanes Detection and Classification

Although the 360° 3D LiDAR frames are rich in information, a single LiDAR frame is still sparse for effective Lane Detection. In order to overcome the problem of data sparsity, we compute a re-emission grid (reflectance of the LiDAR rays) by integrating the multiple LiDAR frames and aligning them based on the vehicle odometry information (for further information about the construction of re-emission map, see [12]). The grid map covers $70.2 \times 70.2$ meters of the world around the car. The grid cell has the size of $0.2 \times 0.2$ meters. This results in an odd size map of dimensions $351 \times 351$, and thus this odd size ensures that the ego vehicle is always virtually placed in the center. The position of the car within the local map is updated using the car motion model, and the local map is further translated accordingly. In order to simplify further processing of data, we create a saliency image of dimension 351 × 351 pixels which is the surrogate representation of the re-emission grid. Each pixel in the saliency image thus corresponds to the respective cell within the re-emission grid, and the mean reflective value within each cell is further stored as the equivalent pixel intensity value. Hence, a 3D point cloud problem is now reformulated as a 2D image processing problem, which simplifies as well as speeds up the computational process [12].

In order to identify the lanes and relevant regions within the saliency image, we binarize it using an image thresholding [13] technique. However, objects which have a varying relative velocity with respect to the ego vehicle introduce noise while thresholding the saliency image. In order to overcome this, we use the approach presented in [14] to compute the probability that the given cell in the re-emission grid lies on a moving object. Those cells which have a higher probability of being classified as a moving object are discarded from the saliency image before being binarized and dilated. The result of the image, shown in Fig. 3a, after thresholding and dilation is shown in Fig. 3b. However, the values resulting out from bushes, guard rails and curbs are still retained at this stage.

In general, the car is between two lane lines, curbs, a curb and a lane line, or some road limits. As the system uses the ego-motion information to create the map and since the car is mostly travelling in a single lane, this information is valuable to extract the lines that restrict a lane (see Fig. 3b).

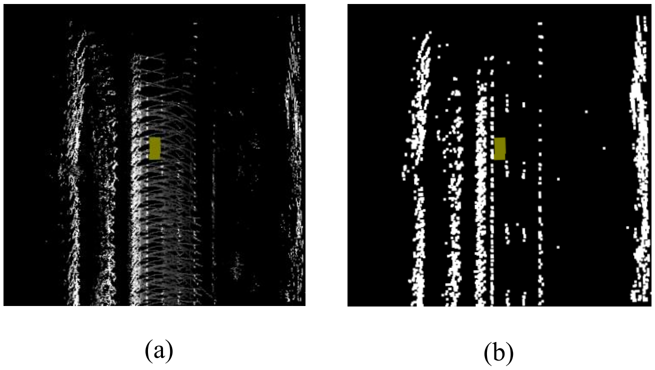


Fig. 3: (a) shows a re-emission grid map. Here, the gray scale part represents the cell touched by the 3D LiDAR. The yellow box illustrates the position of the car related to map. (b) shows the map after applying thresholding and dilation operations. The white points on the map represent possible lane candidates. The lanes themselves are also visible. The noise needs to be removed in order to detect the ego and side lanes. The yellow box in (a) and (b) represents the car.

Extending the lateral of the car 1.7 meters for 15 previous positions of car, also, considering 15 positions ahead, the masks are built. Fig. 4 presents the ego-lane masks (Fig. 4a) used to detect lines of the lane on binarized map in Fig. 4b. After the use of the mask to extract the points of the lines from the map (Fig. 4c), a quadratic regression finds the polynomial coefficients which fits with each ego line. Fig. 4d shows the curve of polynomial output.

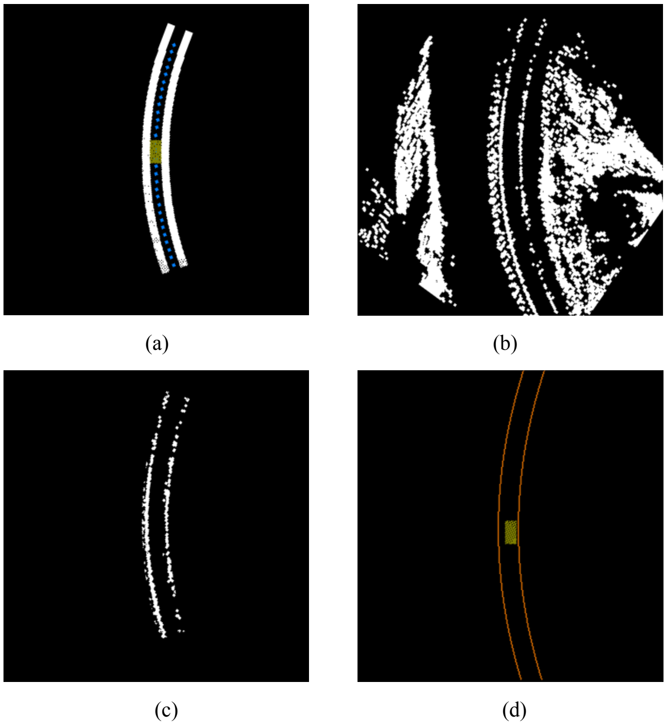


Fig. 4: Sequence to compute the lines of the ego-lane. The mask (a) is applied on binarized map (b) to extract the points of the points of the lines (c). From the ego-lane point a regression computes the coefficents and the polynomial curves are drawn in the image (d). The yellow box here represents the car. The blue dots on (a) shows the previous and predicted car pose used to build the masks.

### B. Side-Lanes Detection

The ego-lane detection is very important to have a level of autonomy and safety on cars. However, if the car does not have any method for detecting the side lanes, the autopilot system should rely on the actuation of the driver to perform a lane change. For example, some advanced drive assist needs the driver to turn the indicator lights on to perform lane changes.

In our system, due to the number of false positives in the segmentation (see Fig. 3b), placing the masks expecting side lanes is not feasible since the lane detection system would detect lines on top of obstacles. Therefore, our system measures the width of road extending both sides of car up to 13 meters (approximately width of four lanes. Fig. 5 shows the road width measurement from the current position of the car. The extension goes until it hits a white cell on the image of the map. Then, to detect whether there is a side lane the extension width must be bigger than the size of a lane (3.2 meters for United States). During the side lane detection, the average of 30 meters road width measurement is taken extending the car side based on previous and predicted poses of car. The same masks are used to extract the lines of both sides, just by translating the mask one lane's size for the side which has a side lane.

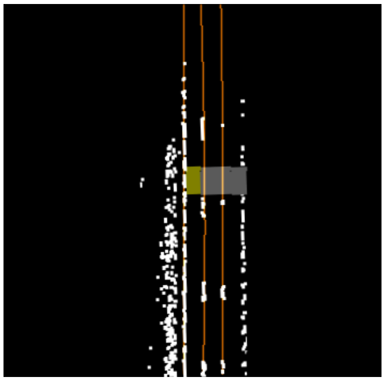


Fig. 5: The road width (grey) measurement. Here it is possible to see that the side-lane on the right side of car was detected but not on the left. With noise on left of the car, it was possible to detect a lane. However, by measuring road width, the system was robust enough to prevent this.

## III. Experimental Setup

This section presents the autonomous car platform used to collect the dataset. We describe briefly the LiDAR lane detection framework and libraries used to implement it. Also, the metrics used and the methodology for evaluating the lane detection is shown.

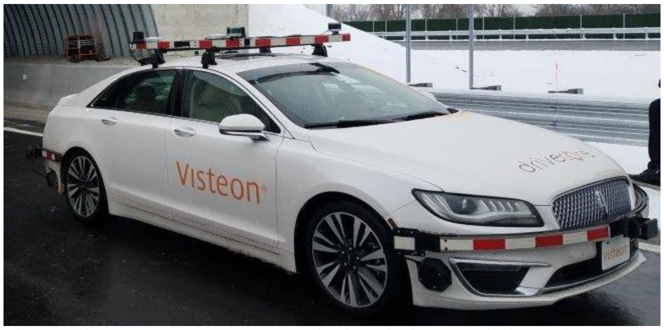


Fig. 6: Visteon's autonomous car robotics platform. Velodyne LiDARs are mounted on the top of the car.

### *A. Autonomous Car Robotics Platform*

To collect the data to evaluate the LiDAR based lane detection, we used the Visteon's autonomous car robotics platform. Fig. 6 shows the picture of our 2017 Lincoln MKZ Hybrid equipped with the many sensors. The sensors used in this work are two VLP 16 Velodyne Pucks and IMU to get the angular and linear velocities.

### *B. LiDAR Lane Detection Framework*

The robotic platform has Linux Ubuntu 16.04 with a real-time kernel. All the software components are written using ROS Kinetic. Therefore, the LiDAR synchronization is a separate node from a lane detection. This node subscribes to the messages from two 3D-LiDARs and ego-motion node and fuse both point-clouds considering the car movement. All the safety treatment for frame missing or sensor absence is treated. It publishes a single message with two point-clouds synchronized by the time-stamp of the last one to arrive. To apply all the transformations, this node is implemented using PCL (Point Cloud Library) [15] and Eigen.

The LiDAR lane detection system receives the messages from point-clouds and also the ego-motion, which has the linear and angular velocities. The car motion model is applied to have a local pose of the car. The car is placed on the map using the local pose and every ray position according to map is computed. This is the mapping phase of our algorithm and outputs the map shown in Fig. 3a. This grid map is converted into an OpenCV image to be processed [13]. After extracting all the points belonging to each lane, a regression implemented using Eigen generates the coefficients of second order polynomial. Those coefficients are published to a node that would compute the curvature of the lanes for lane keeping planner.

### *C. Dataset*

The datasets were collected using our in house autonomous (see Fig. 6) car platform driven by a subject. They were recorded in the middle of the day in Detroit area, United States of America (see Fig. 7). For evaluating the LiDAR based lane detection, four ROS bags were collected (Bag 1 - Fig. 7(1), Bag 2 - Fig. 7(2), Bag 3 - Fig. 7(3), Bag 4 - Fig. 7(4)). The courses were mostly on a highway since we were looking to test the side lane detection robustness. As most of the time on highway the car was driving close to the speed limit of the highway, the dataset itself is very challenging. It could be seen in Fig. 3a by the space between the scans. Table I shows the recording time, the path size, and the average speed for all four ROS bags used.

Table I: The four dataset time of recording in seconds, path kilometers and Average Speed kilometers per hour.

| Dataset | Time(s) | Size (km) | Average Speed (km/h) |
|---|---|---|---|
| Bag 1 | 356 | 9.6 | 97.2 |
| Bag 2 | 132 | 4.4 | 119.6 |
| Bag 3 | 413 | 13.3 | 116.3 |
| Bag 4 | 409 | 8.9 | 78.4 |

### *D. Metrics for comparing the detection performance*

In the absence of a ground truth to analyze the performance of our LiDAR based lane detection system, we compare its results with our in house camera based lane detection system. The two metrics used for the comparison are Dice coefficient [16] and Jaccard index [17] (Intersection of Union) [5] [18].

The Dice coefficient also known as F1 score combines both precision and recall in to one metric. Precision and recall are defined as follows

$$Precision = \frac{tp}{tp + fp} \tag{2}$$

$$Recall = \frac{tp}{tp + fn} \tag{3}$$

where $tp$, $fp$ and $fn$ represents true positive, false positive and false negatives respectively.

The Dice coefficient which combines both precision and recall is defined as:

$$Dice = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{Precision + Recall} \qquad (4)$$

The second metric is Jaccard Index, defined in Eq. (5), commonly known in the field of computer vision as Intersection of Union (IoU).

$$Jaccard = \frac{tp}{tp + fp + fn} \qquad (5)$$

Both, Dice coefficient and Jaccard Index are positively correlated but quantitatively Jaccard Index penalizes an instance of a bad classification more than Dice coefficient. Consider a case where we have two classifiers A and B and most of inferences of classifier A are slightly better than classifier B but some of the inferences of classifier A are significantly worse than classifier B, then it might be the case that on average, classifier A has better Dice coefficient score and classifier B has better Jaccard Index. So, Dice coefficient is a metric which quantifies average performance whereas Jaccard Index is a metric which quantifies worst case performance of a classifier.

## IV. Experimental Results

The evaluation of our lane detection was performed on four datasets (ROS bags) collected. To match the frames for comparison, we took the output image of the in-house camera based lane detection solution with the closest time-stamp to LiDAR solution. Even though, the lane detection implemented in this work can run up to 40 Hz, the LiDAR was set up to run at 16 Hz. On the other side, our in-house camera lane detection works close to camera frame rate, 25 Hz. It is worth to mention that quantitative results is computed only for the ego-lanes since our in-house camera based lane detection does not detect the side lanes yet.

### A. Quantitative Results

The quantitative results are produced by comparing our LiDAR based lane detection with our in-house camera based lane detection. Proposed LiDAR based solution has the advantage to use the lanes behind the car, which could help to detect collision in the back of the car. However, ground truth is just up to front of the car. Thus, to perform the comparison, the lanes on the back side detected by LiDAR based lane detection are removed in order to overlay both results. Four ROS bags were used to evaluate the lane detection performance. Table II shows the average value for Dice coefficient and Jaccard index. Fig. 8 and Fig. 9 show the resulting histogram of Dice coefficient and Jaccard Index for the respective four dataset bags.

TABLE II: DICE COEFFICIENT AND JACCARD INDEX OF THE LIDAR BASED LANE DETECTION FOR FOUR DIFFERENT ROS BAGS COMPARING AGAINST THE IN-HOUSE CAMERA BASED LANE DETECTION.

| Dataset | Dice Coefficient (%) | Jaccard Index (%) |
|---|---|---|
| Bag 1 | 77.80 | 66.38 |
| Bag 2 | 55.90 | 43.22 |
| Bag 3 | 71.33 | 58.66 |
| Bag 4 | 66.23 | 54.82 |

### B. Qualitative Results

In this section we present the qualitative results of our approach. Fig. 10 shows visually the positive matches of lines on both camera and LiDAR based lane detection methods. Fig. 11 expose some of the failure cases where the camera works and the LiDAR fails, camera fails and the LiDAR works and both fails.

Besides, Fig. 12 shows a sequence of images and maps with the outcomes of the LiDAR based lane detection working even with a truck blinding the LiDAR view.

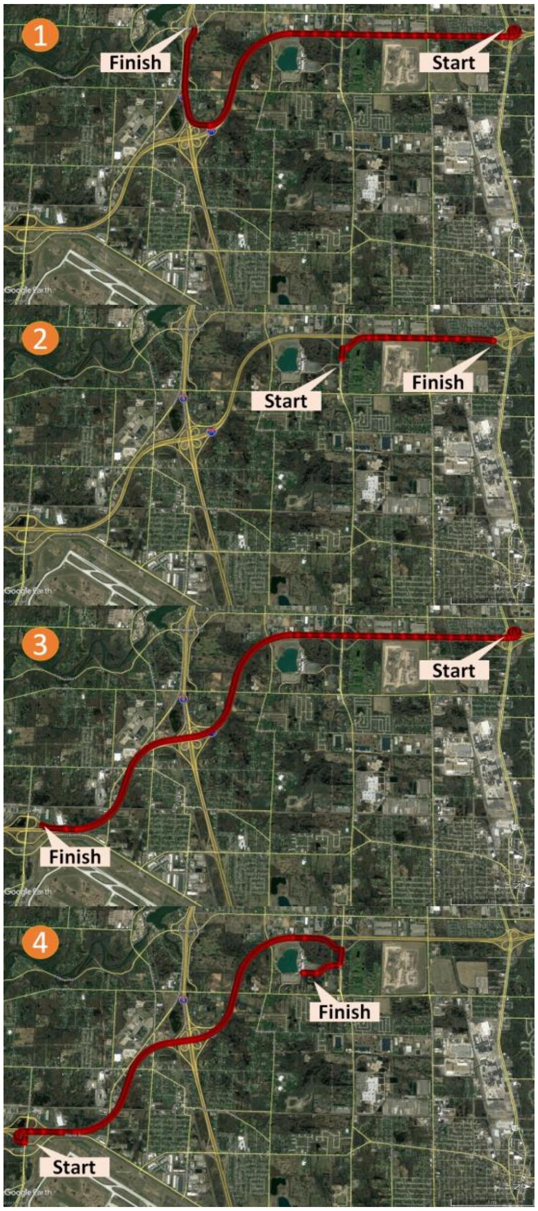
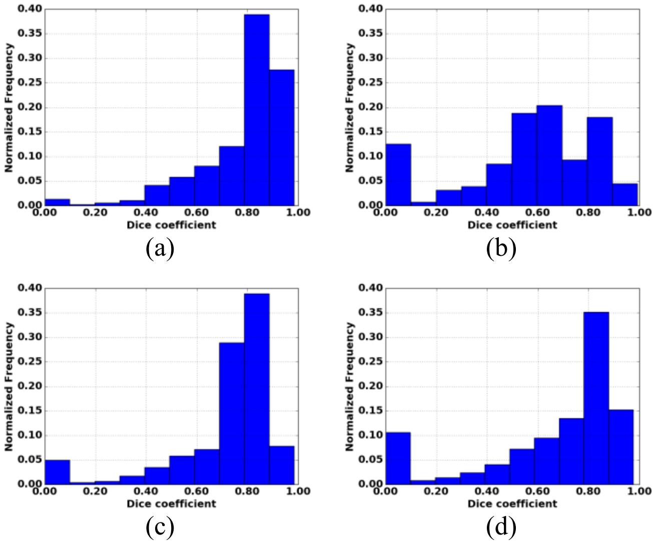
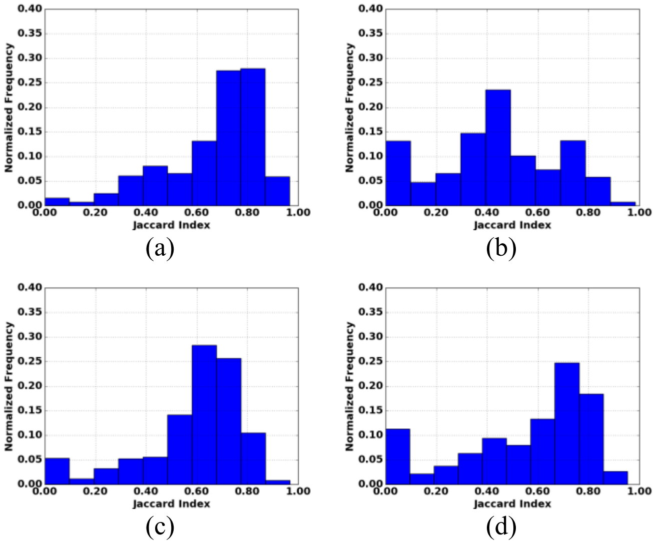


*Fig. 7:* Google-Earth picture for all four ROS bags the dataset paths (red). Thus we recorded data from four different routes, which has slightly different road and driving conditions.



(a) (b) (c) (d)

Fig. 8: Dice coefficient histogram for the four dataset bags (i.e., (a) Bag 1, (b) Bag 2, (c) Bag 3, (d) Bag 4). The frequency of occurrence is normalized in order to sum heights of all bars equals to one.



(a) (b) (c) (d)

Fig. 9: Jaccard Index histogram for the four dataset bags (i.e., (a) Bag 1, (b) Bag 2, (c) Bag 3, (d) Bag 4). The frequency of occurrence is normalized so that sum of heights of all bars equals to one.

## V. Discussion

Evaluating qualitatively the LiDAR based lane detection by the videos posted (https://youtu.be/n0v9xOSIQww), Fig. 10 and Fig. 12, we can see a promising future for LiDAR in this role. Furthermore, while the camera is completely blind to the side lanes in the case shows in Fig. 12, our solution uses the prior information which avoid the losing side lanes perception in such a case.

However, quantitative results are difficult to get since the LiDAR now-a-days spins mechanically to have 360° imagery of the environment. On top of that, the vertical sparsity of the 3D LiDAR makes the integration of more than one 360° scan to extract valuable and meaningful information from the sensor. Thus, the annotation of datasets becomes harder. So, we have used our in-house camera based lane detection to confront results.

Since the side lane detection on the camera solution has some issues when the view is blind by a vehicle or limited field of view of the camera, just the ego-lane detection has been compared. Table II illustrate the correlation of the methods by the Dice coefficient and Jaccard Index. The histograms in Fig. 8 and Fig. 9 emphasize this correlation. According to those graphs, more than 70% of the time in the Bag 1, 3 and 4 the lines were superimposed. On the other hand, in Fig. 8b it is noticeable that miss match happen more often. This happened since in this bag car stopped in a lot intersections and the LiDAR method needs car dynamic to build the masks. Moreover, the course has narrow road with two lanes separated by a line and limited by two curbs, and the camera based lane detection is not trained to detect curbs as a lane limits.

Besides, the outcomes difference could be explained by the difference in the lane detection methods. The methods behave in different ways during lane change (shown by Fig. 11 third row) as the assumption to build the masks for detecting the lanes using LiDARs is that the car is travelling most of the time inside a lane. Thus, the metrics might go to zero in some images during lane changing.

## VI. Conclusion

We have developed a LiDAR based lane detection which is able to detect the ego and side lanes. The system integrates several frames from different 3D LiDARs in a local grid map using the vehicle motion model. The map is infrared reflectance image of the environment around the car. This image is binarized by thresholding operations to accentuate the line of the lanes. The lines of the lanes are extracted by applying a mask built using 15 previous positions of the robot on the map, and predicting 15 positions ahead with time resolution of 0.1 seconds. Therefore, the polynomial is taken using a quadratic regression.

Using simple algorithms of image processing, the final solution was tested on a single ARM core of the NVIDIA Drive PX2 without the need for GPUs, and achieved a frame rate of 40 Hz. In the absence of ground truth, we evaluated our LiDAR based lane detection system against our in house camera based lane detection system. The system was evaluated for bags of data, and the highest average Dice coefficient in a bag was 77.80%, and the highest Jaccard index was 66.38%.

As future work, we will add more features to the LiDAR based lane detection in order to avoid lane jumping during lane change. Also, we are planning to create a calibration system where the system could adapt with different lane width.

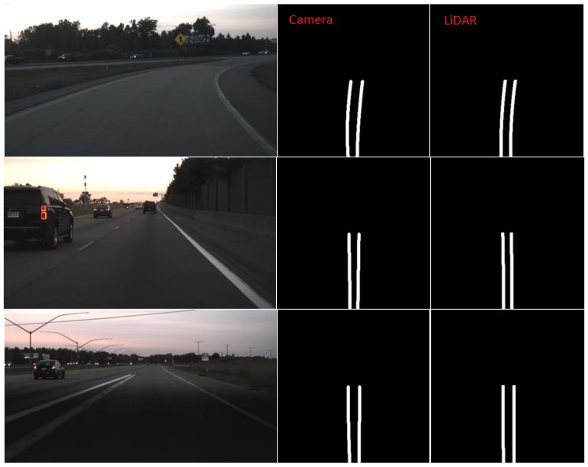


Fig. 10: Visual results of positive cases of LiDAR and camera based lane detection. First column shows the input images. Second column shows the result of camera based lane detection and the third column demonstrates the results of LiDAR based lane detection and Table III gives corresponding Dice coefficient and Jaccard Index.

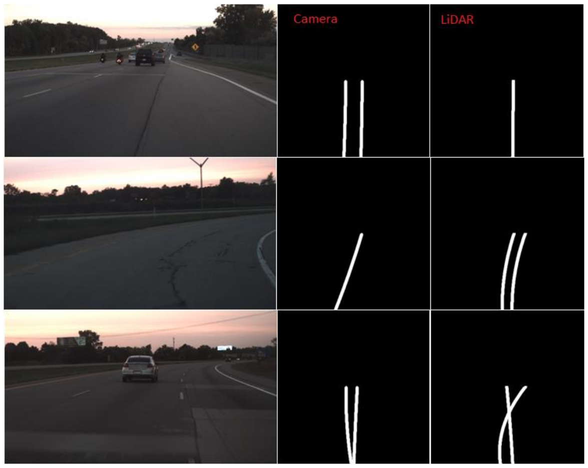


Fig. 11: Visual results of negative cases of LiDAR and camera based lane detection. First row shows the scenario where camera lane detection works but LiDAR lane detection fails. Second row shows the results where LiDAR lane detection works but camera lane detection fails. The third column shows the scenario where both methods fails. Table IV gives corresponding Dice coefficient and Jaccard Index.

TABLE III: DICE COEFFICIENT AND JACCARD INDEX OF THE LIDAR BASED LANE DETECTION FROM THE IMAGE 1 TO 3 OF FIG. *10*.

| Dataset | Dice Coefficient (%) | Jaccard Index (%) |
|---|---|---|
| Image 1 | 80.20 | 66.90 |
| Image 2 | 83.10 | 71.09 |
| Image 3 | 90.88 | 83.20 |

TABLE IV: DICE COEFFICIENT AND JACCARD INDEX OF THE LIDAR BASED LANE DETECTION FROM THE IMAGE 1 TO 3 OF FIG. *11*.

| Dataset | Dice Coefficient (%) | Jaccard Index (%) |
|---|---|---|
| Image 1 | 24.82 | 14.17 |
| Image 2 | 22.00 | 12.30 |
| Image 3 | 25.77 | 14.79 |

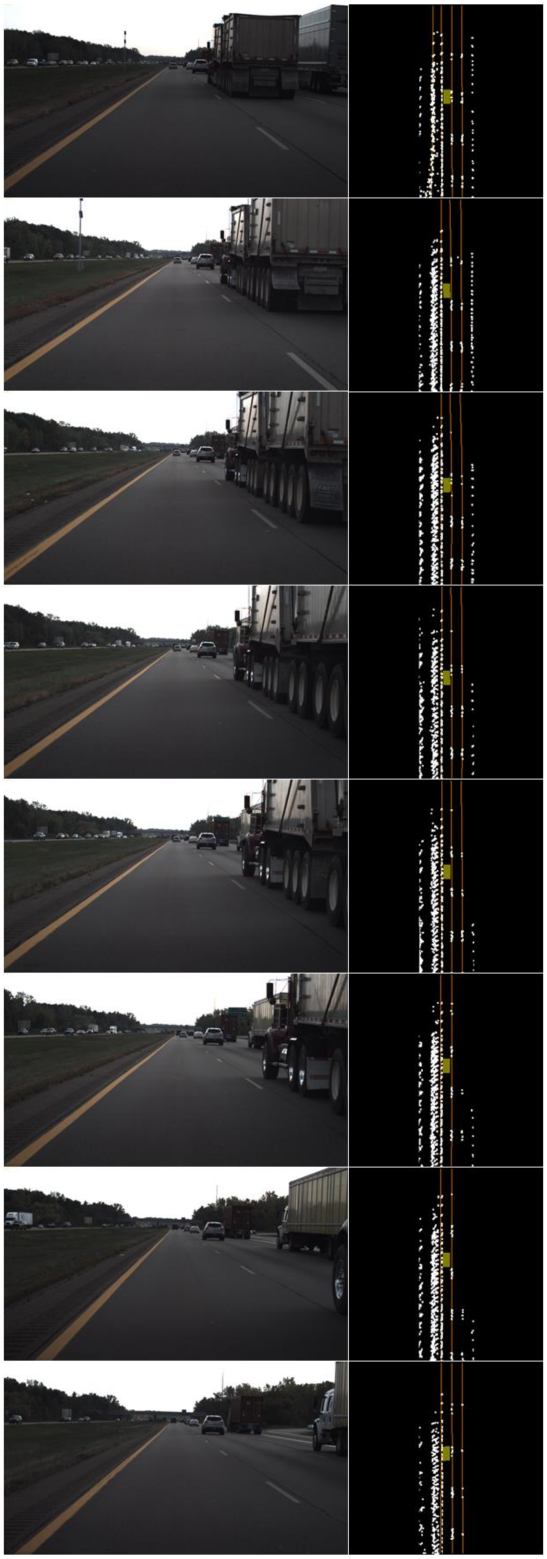


Fig. 12: Sequence outcomes of the LiDAR based lane detection detecting the side-lanes even with a truck blinding the LiDAR view. At some point, it missed the side lane shown in the rows 6 and 7, but it recovered just thereafter.

## REFERENCES

[1] M. Buehler, K. Iagnemma and S. Singh, The 2005 DARPA grand challenge: the great robot race, vol. 36, Springer Science & Business Media, 2007.

[2] M. Buehler, K. Iagnemma and S. Singh, traffic, The DARPA urban challenge: autonomous vehicles in city, vol. 56, springer, 2009.

[3] R. Guidolini, C. Badue, M. Berger, L. Veronese and A. F. De Souza, "A simple yet effective obstacle avoider for the IARA autonomous car," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, 2016.

[4] V. Cardoso, J. Oliveira, T. Teixeira, C. Badue, F. Mutz, T. Oliveira-Santos, L. Veronese and A. F. D. Souza, "A Model-Predictive Motion Planner for the IARA autonomous car," in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller and J. Zhang, "End to end learning for self-driving cars," *arXiv preprint ,* vol. arXiv:1604.07316, no. self-driving cars, p. 2016.

[6] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt and C. G. Keller, "Making Bertha drive—An autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine,* vol. 6, pp. 8--20, 2014.

[7] A. Jain, J. Jiang, J. Lui, R. D'Souza and S. Limarta, "Google Self-driving Cars," Google, 2013.

[8] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. a. M. T. Rajpurkar and R. Cheng-Yue, "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1504.01716,* 2015.

[9] C. Chen, A. Seff, A. Kornhauser and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015.

[10] S. Kammel and B. Pitzer, "Lidar-based lane marker detection and mapping," in *Intelligent Vehicles Symposium, 2008 IEEE*, 2008.

[11] A. B. Hillel, R. Lerner, D. Levi and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications,* vol. 25, pp. 727--745, 2014.

[12] F. Mutz, L. Veronese, T. Oliveira-Santos, E. de Aguiar, F. Cheein and A. De Souza, "Large-scale mapping in complex field scenarios using an autonomous car," *Elsevier,* vol. 46, no. Expert Systems with Applications, pp. 439--462, 2016.

[13] G. Bradski and A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library, O'Reilly Media, Inc, 2008.

[14] L. Veronese, J. Guivant and A. D. Souza, "Improved Global Urban Localization Based on Road Maps and 3D Detection of Road Intersections," in *Australasian Conference on Robotics and Automation*, Canberra, 2015.

[15] R. Rusu and S. Cousins, "3D is here: Point cloud library (pcl)," in *IEEE International Conference on Robotics and automation (ICRA)*, 2011.

[16] L. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology,* vol. 26, pp. 297-302, 1945.

[17] P. Sneath, "Some Thoughts on Bacterial Classification," *Microbiology,* vol. 17, no. Microbiology, pp. 184-200, 1957.

[18] V. Thada and V. Jaglan, "Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm," *International Journal of Innovations in Engineering and Technology,* vol. 2, pp. 202-205, 2013.