2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Deep Multi-Sensor Lane Detection

Min Bai*,1,2 Gellert Mattyus*,1 Namdar Homayounfar1,2 Shenlong Wang1,2
Shrinidhi Kowshika Lakshmikanth1 Raquel Urtasun1,2

***Abstract*— Reliable and accurate lane detection has been a long-standing problem in the field of autonomous driving. In recent years, many approaches have been developed that use images (or videos) as input and reason in image space. In this paper we argue that accurate image estimates do not translate to precise 3D lane boundaries, which are the input required by modern motion planning algorithms. To address this issue, we propose a novel deep neural network that takes advantage of both LiDAR and camera sensors and produces very accurate estimates directly in 3D space. We demonstrate the performance of our approach on both highways and in cities, and show very accurate estimates in complex scenarios such as heavy traffic (which produces occlusion), fork, merges and intersections.**

I. Introduction

Lane detection is one of the fundamental problems in autonomous driving. To drive safely, the vehicle must reliably detect the boundaries of its current lane for accurate localization, while also estimating the nearby lanes for maneuvering. Accurate detections of lane regions can help reduce ambiguities when detecting and tracking other traffic participants. Furthermore, it is an essential component for the automatic creation of high definition maps, as well as error checking and change detection in existing maps.

Most approaches to lane detection take an image (or a video) as input and estimate the lane boundaries in image space [1], [2], [3], [4], [5], [6]. In this paper, we argue that accurate results in the camera view may not imply accurate estimates in 3D space. This is due to the fact that the perspective projection of the camera causes the spatial resolution in 3D to decrease drastically with distance from the camera. This is a big issue for modern self driving vehicles as motion planners require a birds eye view representation of the lane topology. We refer the reader to Fig. 2 for an illustration of this problem. The figure shows sample outputs of the state-of-the-art camera based lane detection approach of [2]. Despite very accurate results in camera perspective, the results in 3D are not very accurate.

Several approaches that directly use 3D sensors such as LiDAR have been proposed [7], [8], [9], [10], [11], [12]. Although LiDAR gives an unambiguous measurement of 3D points, it is spatially much more sparse when compare to images, especially at long-range. Consequently, many of the proposed methods relied on handcrafted features and strong assumptions, *e.g.* fitting parabolic curves. Additionally, they often fall short in reliability, especially in less ideal scenarios like heavy occlusion. While camera-based techniques have been fairly extensively explored, the possibility to exploit the rapid progress in deep learning to boost the accuracy and reliability of lane detectors using LiDAR or multi-sensor input remains open.

In this paper we propose a novel deep neural network that takes advantage of both LiDAR and camera sensors and produces very accurate estimates directly in 3D space. Our network can be learned end-to-end, and can produce reliable estimates with any combination of sensors. Importantly, our approach is very efficient and runs in as little as 70ms on a single Titan Xp GPU when using all sensors. We demonstrate the effectiveness of our approach on two large scale real world datasets containing both highway and city scenes, and show significant improvements over the state-of-the-art.

II. Related Work

This section discusses the various techniques that have been proposed to tackle the lane detection task. The techniques are grouped by their themes.

a) Camera view lane detection: A commonly seen family of solutions reason in the first person perspective of a vehicle. This includes a number of traditional computer vision techniques based on feature extraction using handcrafted methods to identify likely locations of lane markings. Such methods include the use of various edge detection filters [1], corner detection features [13], Haar-like features [14], and clustering [15]. This is then often followed by line or spline fitting with techniques such as RANSAC to yield final lane detections. Some techniques further use conditional random fields to refine the outputs [6]. However, these techniques are often held back by their inability to reason in abstract domains in the presence of heavy occlusion by other traffic participants, varying lighting conditions between day and night scenes, and weather conditions. Additionally, due to the perspective transformation by the camera, far away road and lane markings are simultaneously heavily distorted and reduced in resolution.

Recent advances in convolutional neural networks (CNN) have led to a drastic increase in performance in various 2D computer vision tasks. Many methods have been proposed that leverage these powerful models for feature extraction or direct lane detection [2], [3], [4], [5].

Some techniques have been proposed which attempt to detect lanes in the camera view, but additionally use regularities in the 3D space to guide the detections. For example, [16] exploits the fact that lanes are mostly parallel lines in 3D and

[1] MB, GM, NH, SW, SKL, and RU is with Uber Advanced Technologies Group, {mbai3, gmattyus, namdar, slwang, klshrinidhi, urtasun}@uber.com
[2] MB, NH, SW, RU is with University of Toronto, {mbai, namdar, slwang, urtasun}@cs.toronto.edu

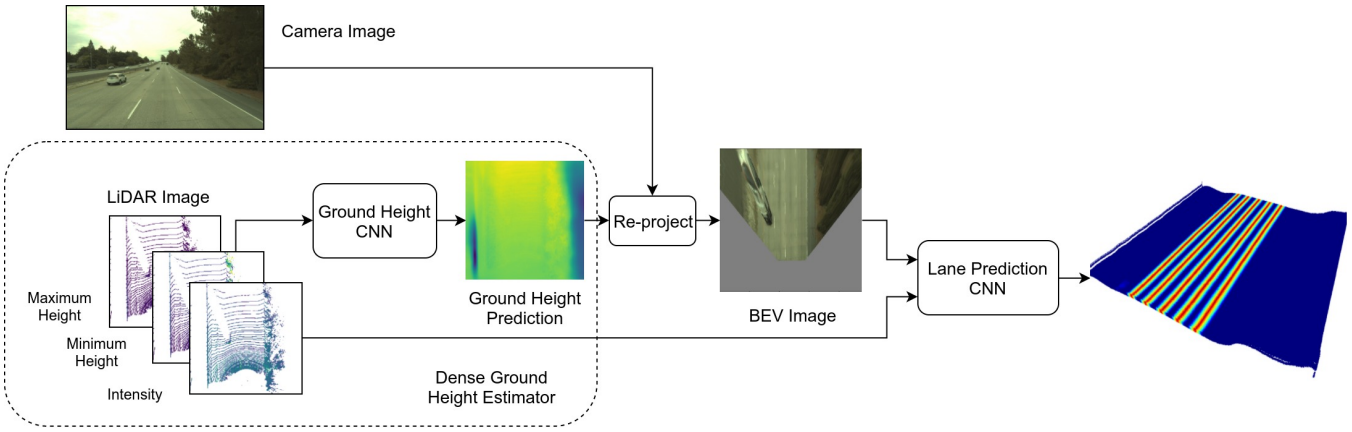

Fig. 1. Complete network architecture. Our model takes as input information extracted from LiDAR sweeps, and predicts a dense ground height. The input RGB camera image is projected onto the dense ground, and is combined with the LiDAR information to produce lane detections in overhead view 3D view.

converge at a vanishing point in the projected image. They use the 2D detections of these vanishing points to further guide their CNN-based lane detection model. Additionally, [17] explored an interesting framework which uses transfer learning methods to improve lane detection performance on different datasets.

b) Overhead view lane detection: As will be shown later in our work, reasoning about lanes in the overhead view domain has significant advantages. This setting has previously been explored by [18], where the authors use a siamese CNN to simultaneously reason about both overhead and camera view input. An earlier work by [19] uses perspective mapping to project camera images into an assumed fixed ground plane and a relatively simple neural network model for the related task of road symbol detection. There, they demonstrated the benefits of reasoning in the overhead view as it reduces the perspective distortion of far away road symbols. This scheme is followed by other authors, including [20]. However, these schemes neglect the natural swaying of the ego-vehicle due to its suspension system, as well as inherent sloping of the road. This limits the spatial accuracy of their detections. In contrast, we propose an end-to-end trained deep learning model that predicts an accurate ground height estimation as an intermediate step onto which the camera image is projected.

c) LiDAR-based lane detection: Several techniques have been proposed using LiDAR measurements as the input [7], [8], [9], [10]. However, these techniques largely do not leverage the recent advances in deep learning. Consequently, they either assign each point a label resulting in only sparse labels, or using some underlying assumption, such as parabolic curve to fit a lane model. Other works have successfully applied CNNs to the task of road segmentation, especially in the overhead view [21]. Our work takes this one step further to show that information from LiDAR is sufficient for dense and accurate lane boundary detections as well.

d) Multi-sensor lane detection: Previous works also exploit the use of multiple sensors to boost the performance of lane detection and tracking [11], [12]. However, lane perception only uses the cameras, while LiDAR is used for supplementary tasks such as obstacle masking and curb fitting. Unlike these methods, our approach aggregates information from both sensors to detect lanes in the same output space.

III. Multi-model End-to-End Lane Detection

In this section, we propose a new model that produces accurate and reliable lane boundaries in 3D space. Our model combines the strength of camera's dense observations as well as LiDAR's unambiguous 3D measurements. In the following, we first define the input and output parameterization, followed by a description of the architecture of our proposed end-to-end learnable model. We then discuss each individual component in detail as well as how to train our model.

A. Parameterization of the Input and Output

Our approach is inspired by the observation that performing lane detection in image space can result in large inaccuracies in 3D (especially at long range). To address this issue, we phrase the problem as a dense pixel-wise prediction task in 3D vehicle coordinates, regardless of the type of sensor input employed. This enables the lane detection to directly deliver lane boundaries in the same space in which the motion planning and control system operate.

Most existing pixel-wise CNN-based approaches (e.g. [2]) attempt to predict whether or not a given pixel belongs to a lane boundary. However, they disregard the distinction between cases where the detected lane boundary is slightly shifted compared to the ground truth, and cases where the detection is completely spurious or missing. To address this problem, we phrase the output space as the minimum Euclidean distance to the nearest lane boundary at each location. This allows the model to explicitly reason about the relationship of each 3D location and lane boundaries,

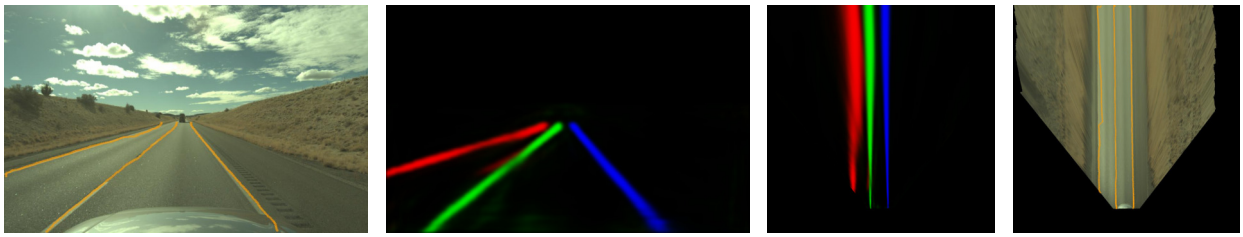

Fig. 2. Illustration of the necessity of reasoning about lane detections in the overhead view. The input camera image overlaid with the predictions from [2] is shown on the left. The lane probability output of the model in camera view is shown next, followed by the re-projection of the probability output into overhead view. The lane detections re-projected into overhead view is on the right. Although the lane detections in camera view appear quite accurate, the shortcomings are clearly visible in the overhead view, as the probability output becomes diffuse with increasing distance, and the detected lane boundaries become less accurate.

providing an output that encapsulates richer information than class labels. Moreover, the loss is more forgiving to small offsets between the prediction and ground truth. To aid the network's learning, we make a small modification to the distance transform. In particular, we eliminate the unbounded regression problem by thresholding the target value, and invert the distance transform by subtracting it from the threshold itself. As a result, the training target is equal to the threshold at the lane boundary, and decays outward to zero.

Our model takes two inputs: a LiDAR point cloud as well as an image of the scene. In particular, we bring the point clouds from five consecutive frames to a common reference frame by correcting for ego-motion. We then rasterize the point clouds to a 3-channel image in Bird's Eye View (BEV) encoding both the intensity and height of the highest point as well as the height of the lowest point in each discretization bin. It is important to note that there may be independently moving objects such as other vehicles in the scene whose motion would not be compensated for. However, by encoding the highest and lowest height in each discretization bin, the network is informed of the regions that contain moving objects. Using this, the network can learn to ignore distracting data, and interpolate where necessary. The second input is the RGB camera image. We refer the reader to Fig. 1 for an illustration of our model. In the next section, we describe our multi-sensor neural network in details.

B. Network Architecture.

To achieve the goal of simultaneously leverage information from the first person view camera image and the BEV LiDAR image, we must align their domain by re-projecting the camera image into BEV as well. Since our camera is calibrated and fixed to the vehicle, the projection of a point $\mathbf{P}_v = (x_v, y_v, z_v)^T$ in the vehicle frame is defined by the projection matrix $C = K[R_v|t_v]$, where K is the camera calibration matrix and R_v, t_v is the rotation and translation from vehicle to the camera coordinate system. Because the vehicle frame is fixed (i.e. x_v, y_v are constants), only the z_v elevation of the ground has to be estimated online to define the projection from ground into the camera.

The simplest solution is to assume that the road is flat and parallel to the vehicle frame, in which case z_v would be a constant as well. Unfortunately, real world roads often have significant slopes over large distances. Moreover, the elevation and pitch of the vehicle relative to the ground is constantly changing due to the vehicle's suspensions. Because of the oblique angle, ground regions far away from the vehicle are covered by very few LiDAR measurements. As well, many LiDAR readings do not come from the ground, but rather various obstacles and other traffic participants. From experimentation, we find that these pose significant difficulties when attempting to use traditional robust estimators such as RANSAC to produce a dense ground height estimate.

Instead, in this paper we take advantage of deep learning and design a network that estimates dense ground from the sparse LiDAR point cloud, which learns to simultaneously ignore objects above the ground, as well as extrapolate to produce a dense ground surface.

Ground Height Estimator: We use a fast convolutional neural net (CNN) based on ResNet50 [22] to predict a dense ground height from the LiDAR input. We reduce the feature dimensions of ResNet50 by a factor of 2 for scales 1 and 2, and a factor of 4 for scales 3 to 5. Additionally, we remove one convolutional block from each of scale 2, 3, and 5. This is followed by large receptive field average pooling inspired by [23]. In particular, we produce three additional feature volumes using pooling with receptive field and stride sizes of 10, 25, and 60, before concatenating the feature volumes with the original output of scale 5. This gives the network the capability to propagate information about the ground height from regions with high LiDAR point density to regions with low density. Next, the feature volume undergoes 3 additional bottleneck residual blocks, before being upsampled using interleaved transposed convolutions and basic residual blocks. The result is a dense ground height image in BEV of size 960×960. This corresponds to a region of 48×48m at 5cm per pixel.

Camera Image Re-projection: We project the camera image to the estimated ground surface using a differentiable warping function [24]. This produces an image warped onto the dense ground height prediction described above. Note that the re-projection process does not explicitly handle 3D occlusions in the ground plane, with pixels of objects above the ground being projected onto the ground surface. However, as markings guiding lane detections are on the

ground plane, the locations of detected lanes will be largely unaffected. The result is a re-projected camera image with the same size as the predicted dense ground image above.

Combining Re-projected Camera Image with LiDAR: we use a second CNN to leverage both the re-projected camera image and the LiDAR input to produce pixel-wise lane detection results. This module takes as input LiDAR data in the same format as the ground height estimator. We design a second CNN that is largely identical to that used in the dense ground height estimation, with the exception that the feature dimension of scales 3 to 5 are only reduced by a factor of 2 relative to the original ResNet50, with no blocks removed. Moreover, we duplicate scales 1 to 3 at the input without weight sharing such that the re-projected camera image and the LiDAR images are passed into separate input branches. The two streams of information are concatenated at scale 4. The output of this network is our distance transform estimates.

C. Model Learning

The parameters Θ of the overall model are optimized by minimizing a combination of the lane detection loss and the ground height estimation loss:

$$l_{\text{model}}(\Theta) = l_{\text{lane}}(\Theta) + \lambda l_{\text{gnd}}(\Theta)$$

The lane detection loss l_{lane} is defined by

$$l_{\text{lane}}(\Theta) = \sum_{p \in \text{Output Image}} \|(\tau - \min\{d_{\text{p,gt}}, \tau\}) - d_{\text{p,pred}}\|^2$$

where $d_{\text{p,gt}}$ and $d_{\text{p,pred}}$ are the ground truth and predicted distance transform values for pixel p, respectively. τ is a threshold used to cap the distance transform values to a range between $[0, \tau]$, as it is unnecessary for the CNN to produce exact distance transform values for regions far away from any lane boundary. Additionally, this is inverted such that the maximum predicted value τ occurs at the lane boundary, and linearly decreases outward to 0. This removes the need for the network to predict a sharp drop-off in distance transform values at the thresholding boundary from τ to 0.

The ground height estimation loss l_{gnd} is defined by

$$l_{\text{gnd}}(\Theta) = \sum_{p \in \text{Output Image}} \|z_{\text{p,gt}} - z_{\text{p,pred}}\|$$

In this case, we select the L1 loss to encourage smoothness in the output.

IV. Experiments

This section discusses the details of our dataset, evaluation metrics and results.

A. Datasets

We evaluate our approach on two datasets collected in North America. The first dataset includes highway traffic scenes with 22073, 2572, and 5240 examples in training, validation, and test set, respectively. The second dataset includes city scenes from a medium sized city, with 16918, 2607, and 5758 examples in the training, validation, and test sets, respectively. The annotations includes the ground truth lane boundaries in both the camera view and overhead view. We use HD maps which contain dense 3D measurements and annotated lane graphs together with a localization system to provide us with dense ground height ground truth for each example. We plan to release this dataset.

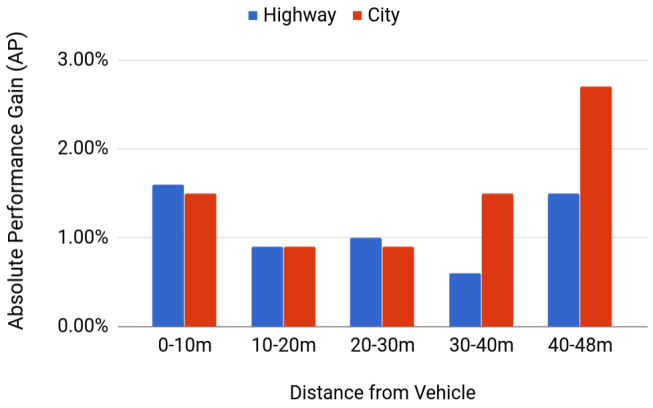

Fig. 3. Difference between the AP at various distances from vehicle of our LiDAR+Camera model versus the LiDAR-only model. It is evident that the LiDAR+Camera model outperforms the LiDAR-only model at all distances, with the advantage further increasing with distance.

B. Experimental Setup

Here, we describe the details of our training process. We set the loss mixing factor $\lambda = 20$. All models are trained using the ADAM [25] optimizer with a learning rate of $1e-4$ and a weight decay of $1e-4$ with a batch size of 8 until convergence as determined by the validation set loss. Additionally, we select the distance transform value threshold of $\tau = 30$ for all models on highway scenes, and reduced it to $\tau = 20$ for all models in city scenes to account for the closer spacing of lane boundaries in the latter scenario. This corresponds to 1.5m and 1.0m in the overhead view, respectively. We further improve the variations in our dataset by augmenting our dataset. We randomly adjust brightness, contrast, saturation and hue of the camera image, and randomly rotate the birds eye view images. To avoid artifacts in the LiDAR image (e.g. interpolate the height between empty and occupied pixels) we first rotate the 3D points prior to rasterization.

C. Metrics

We use two sets of metrics to measure the performance of our model. The first set of metrics directly compares the predicted and ground truth inverted truncated distance transforms over the output prediction by computing the L1 and L2 distances. These metrics naturally penalize both false positive and false negative detections of the lanes at a (overhead) pixel level.

Additionally, we use a simple method to extract a line-based lane boundary representation by thresholding and

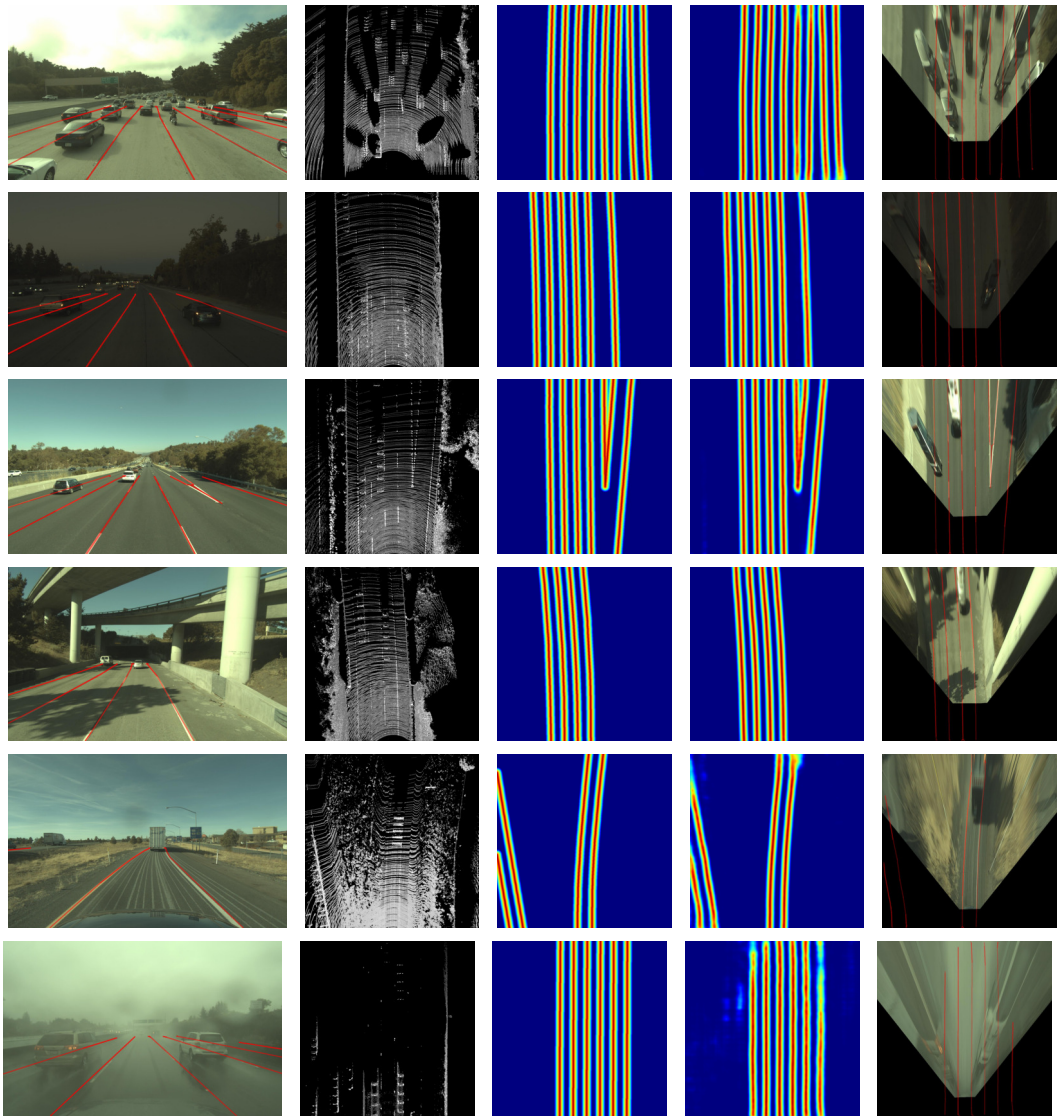

Fig. 4. Output of our method using both LiDAR and camera on highways. From left to right: (1) The detected lane distance transforms are naively thresholded are projected into the camera image using the predicted ground height. Lanes are predicted up to 48m ahead. (2) The LiDAR image. (3) The ground truth lanes. (4) The distance transform output of the model. (5) The camera warped to top-down view using the predicted ground overlaid with the predicted lanes (red). Lane detection can be highly non-trivial: vehicles can occlude the lane markings (rows 1-3), there may be distracting lines (row 5), and rain on both the ground and the camera sensor (last row). However, our model is able to leverage to multiple sources of information to produce high quality lane detections.

binarizing the predictions at a fixed value. The thresholds are selected to be 20 for the highway model, and 15 for the city model, which are 10 px and 5 px away from the lane boundary, respectively. The result is skeletonized via binary erosion, which is then compared to the ground truth lane boundaries by computing the average precision scores at pixel thresholds ranging from 1 to 9. In the overhead image view, this corresponds to physical distances of 5 cm to 45 cm. Moreover, we evaluate the exact precision and recall values at 5 pixel deviation, or 25 cm in the overhead view. This is a stringent requirement, as in comparison the typical lane marking itself is only 15 cm wide.

Finally, we evaluate the correctness of the detected lane topology by finding the number of connected components in the skeletonized prediction, and comparing it with the number of lanes present in the ground truth by computing the absolute difference. This statistic is averaged across all test examples. All results are reported in the test set.

D. Analysis

We present qualitative and quantitative results of our method in this section. The performance of our complete model in highway and city scenes can be found in the third and sixth rows of Table I. Additionally, sample output images can be found in Fig. 4 and Fig. 5. Moreover, we perform ablation studies to examine the factors influencing the performance of our model, as well as comparing with the current state-of-the-art lane detector of [2].

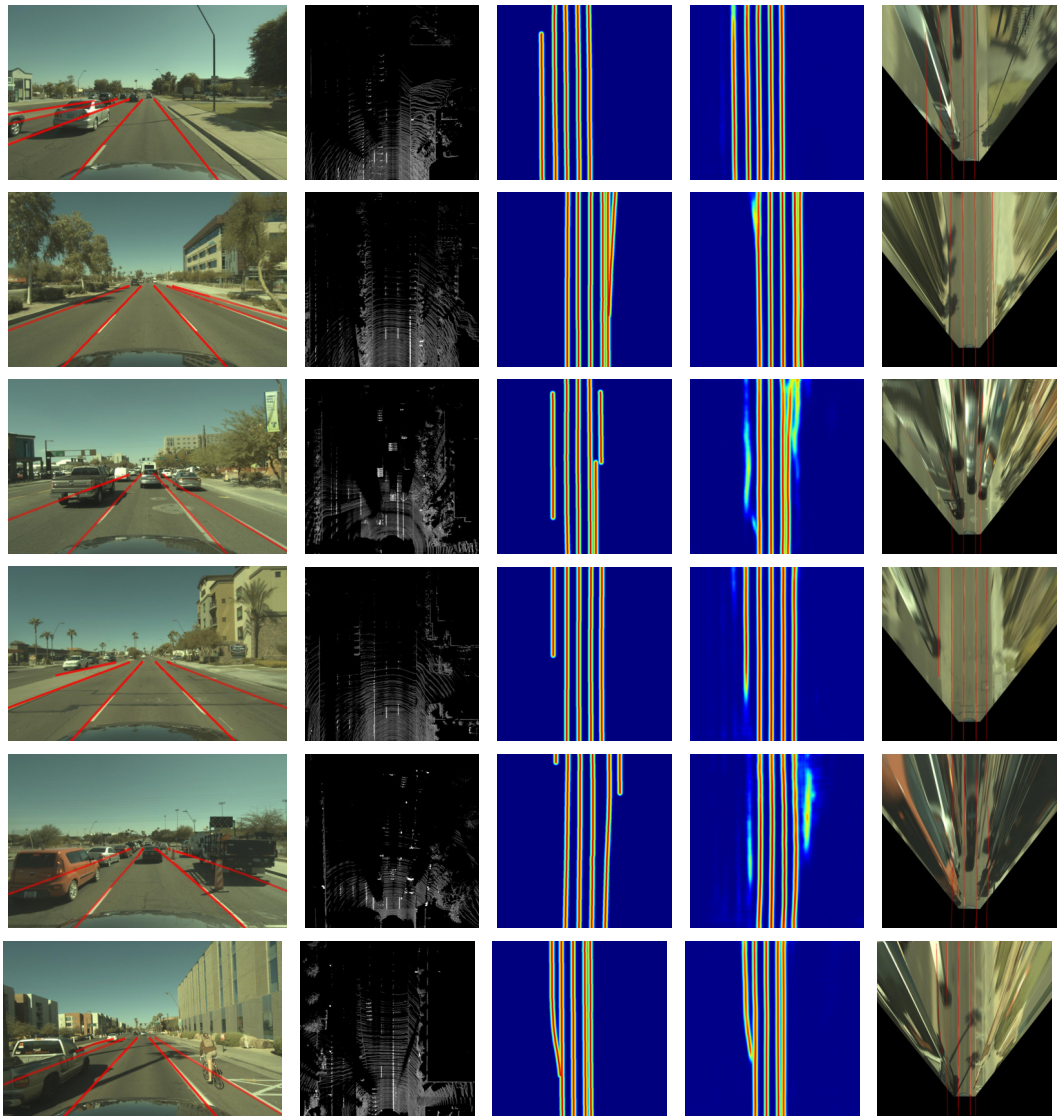

Fig. 5. Output of our method using both LiDAR and camera in urban areas. From left to right: (1) The detected lane distance transforms are naively thresholded are projected into the camera image using the predicted ground height. Lanes are predicted up to 48m ahead. (2) The LiDAR image. (3) The ground truth lanes. (4) The distance transform output of the model. (5) The camera warped to top-down view using the predicted ground overlaid with the predicted lanes (red).

Scenario	Camera	LiDAR	DT L2 (cm^2)	DT L1 (cm)	AP	Pre. @25cm	Rec. @25cm	Top. Mean Dev.
Highway		✓	77.8	2.66	82.9%	95.0%	94.0%	0.337
Highway	✓		110.4	3.35	78.8%	91.3%	89.2%	0.446
Highway	✓	✓	**70.5**	**2.63**	**84.0%**	**95.6%**	**94.3%**	**0.306**
City		✓	111.7	**3.56**	76.7%	88.3%	**79.0%**	1.162
City	✓		130.35	4.68	75.6%	86.3%	72.4%	1.256
City	✓	✓	**109.3**	**3.56**	**78.1%**	**89.3%**	76.9%	**1.053**

TABLE I

COMPARISONS OF TEST SET PERFORMANCE OF LANE DETECTION RESULTS OF USING VARIOUS SENSORS.

a) Sensor Input: we explore the performance impact of using information from the camera image, LiDAR, and the combination of both in Table I. It is clear that the performance of the model that leverages both the LiDAR and camera information achieves the highest performance. This is especially true in the highway setting with faster vehicle motion. As a result, the distant regions still have relatively sparse LiDAR points despite aggregating multiple sweeps, while the much denser camera image is able to somewhat compensate for this. In comparison, the LiDAR-only model performs somewhat worse, while the camera-based model returns the lowest performance.

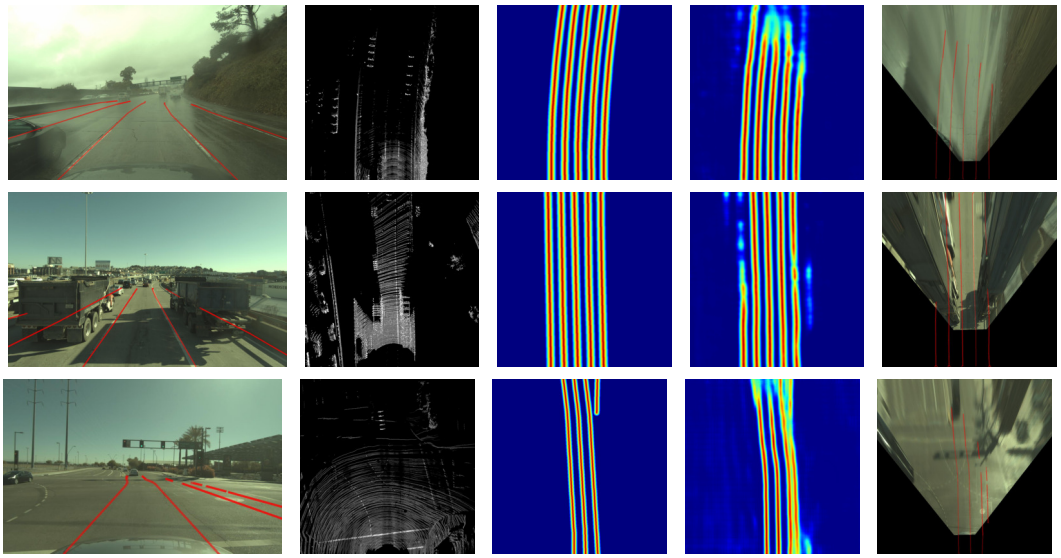

Fig. 6. Failure cases due to rain, occlusions and missing lane markings. From left to right: (1) The detected lane distance transforms are naively thresholded are projected into the camera image using the predicted ground height. Lanes are predicted up to 48m ahead. (2) The LiDAR image. (3) The ground truth lanes. (4) The distance transform output of the model. (5) The camera warped to top-down view using the predicted ground overlaid with the predicted lanes (red).

	AP		Pre @ 25cm		Rec @ 25cm	
Lane Config	SCNN	Ours	SCNN	Ours	SCNN	Ours
2 lanes	64.7%	**85.5%**	75.2%	**96.9%**	76.1%	**95.4%**
3 lanes	48.1%	**78.6%**	54.3%	**90.2%**	45.5%	**83.8%**
4 lanes	67.9%	**84.2%**	78.4%	**96.7%**	52.6%	**95.5%**
5 lanes	66.2%	**82.8%**	76.3%	**95.3%**	57.8%	**94.0%**
6 lanes or more	66.4%	**82.7%**	75.7%	**94.3%**	46.8%	**93.1%**
SCNN compatible	59.5%	**84.7%**	69.1%	**96.3%**	72.3%	**94.3%**

TABLE II

COMPARISON OF TEST SET PERFORMANCE OF LANE DETECTION WITH THE SCNN MODEL AND OUR LIDAR+CAMERA MODEL IN SCENARIOS WITH DIFFERENT NUMBERS OF LANES.

Scenario	Ground Height	DT L2 (cm^2)	DT L1 (cm)	AP	Pre. @25cm	Rec. @25cm	Top. Mean Dev.
Highway	Predicted	**110.4**	**3.35**	**78.8%**	**91.3%**	**89.2%**	0.446
Highway	Ground Truth	112.9	3.55	78.4%	91.0%	88.4%	**0.391**
City	Predicted	130.4	4.68	**75.6%**	**86.3%**	72.4%	1.256
City	Ground Truth	**124.8**	**3.86**	74.7%	85.7%	**73.4%**	**1.210**

TABLE III

COMPARISONS OF LANE DETECTION RESULTS USING ONLY RE-PROJECTED CAMERA IMAGE ONTO GROUND TRUTH GROUND HEIGHT VS PREDICTED GROUND HEIGHT.

b) Comparison with state of the art: we compare the lane detection results using the SCNN model [2] with ours, with the results in Table II. In particular, we train the SCNN model using the authors' provided code on the camera images (as by their design) to produce lane detections in camera space. These detections are then re-projected into overhead view using the ground truth dense ground height and evaluated. While SCNN can only detect the ego lane and the neighboring left and right lanes, our model is able to detect all visible lanes within the 48 meter square. As such, the recall metric for SCNN is low in situations with more lanes. We showcase the performance difference across scenes with varying number of lanes. Because the SCNN detection receives only the image as input which cannot see the ground immediately next to the ego-vehicle, we ignore lanes within 15m of the ego-vehicle for both methods for a fair comparison. Finally, the last row of the table shows the methods evaluated over only scenes that contain an ego lane and at most one lane to the left and right, which is fully within the stated capabilities of SCNN. Our method signifiantly outperforms SCNN in all situations, further validating the effectiveness of our 3D reasoning.

c) Evaluation over Distance: because of the foreshortening effect, the density of measurement points falling onto the ground falls off rapidly with the distance from the vehicle. Since LiDAR sweeps have much lower resolution

than a camera image, this effect is more pronounced in the former case. To further analyze the benefit of using both camera and LiDAR data compared with using only the latter, we plot the relative average precision of the two models versus distance from the vehicle, as shown in Fig. 3. It is clear that in both scenarios, the performance of the former model exceeds that of the latter. This performance gap is especially large at the longest distance bin, where the number of LiDAR points on the ground is very sparse even with sweep aggregation. This suggests that intelligently combining the information from both sensors will provide the best lane detections.

d) Ground height prediction: our model is able to predict the ground height onto which the camera images are projected. In Table III we explore the performance gains possible if the accuracy of ground height estimation is improved by substituting the predictions with ground truth. For this experiment, we only provide the LiDAR BEV image to the ground height estimator so that the model can only perform lane detection using the re-projected camera image. We see that the performance is very similar when using estimated and perfect ground.

e) Qualitative Analysis: in Fig. 4 and Fig. 5, we see a number of examples where the lane detections are very accurate. The high level of performance is achieved in a variety of scenarios including with complex lane geometries (large number of lanes or merges and exits), as well as occlusions by other vehicles. Moreover, the alignment of the lanes in both the camera view and in BEV are seen to be of high quality. The performance of the lane detector in adverse scenarios in Fig. 4 such as darkness, rain, and fog suggests that the model is able to perform in a variety of situations. Finally, the smoothness of the predicted lane boundaries is noteworthy, and highly beneficial to the consumers of the detection results.

f) Failure modes: despite the generally very accurate lane detections, there are a few failure modes of our model. These can be seen in Fig. 6. The first row shows a case where the heavy rain causes significantly reduced visibility, especially at a distance. In the second row, large vehicles block the view of the ground for both sensors, showing the negative impact of heavy occlusions. Finally, the third image shows a case where the lane boundaries do not have actual paint in reality. In these scenarios, more advanced models are required to infer the virtual lane boundaries.

V. Conclusion

In this paper we have argued that accurate image estimates do not translate to precise 3D lane boundaries, which are the input required by modern motion planning algorithms. To address this issue, we have proposed a novel deep neural network that takes advantage of both LiDAR and camera sensors and produces very accurate estimates directly in 3D space. We have demonstrated the performance of our approach on two challenging real world datasets containing both highway and city scenes, showing its superiority when compared to the state of the art. In the future we plan to reason about lane attributes such as lane types and other road elements such as crosswalks.

References

[1] M. Haloi and D. B. Jayagopi, "A robust lane detection and departure warning system," in *IV*, 2015.

[2] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *AAAI*, 2018.

[3] J. Kim and C. Park, "End-to-end ego lane estimation based on sequential transfer learning for self-driving cars," in *CVPR Workshop*, 2017.

[4] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, Mykhaylo, Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A. Y. Ng, "An empirical evaluation of deep learning on highway driving," *arXiv*, 2015.

[5] O. Bailo, S. Lee, F. Rameau, J. S. Yoon, and I. S. Kweon, "Robust road marking detection and recognition using density-based grouping and machine learning techniques," in *WACV*, 2017.

[6] J. Hur, S.-N. Kang, and S.-W. Seo, "Multi-lane detection in urban driving environments using conditional random fields," in *IV*, 2013.

[7] S. Kammel and B. Pitzer, "Lidar-based lane marker detection and mapping," in *IV*, 2008.

[8] P. Lindner, E. Richter, G. W. an Kiyozaku Takagi, and A. Isogai, "Multi-channel lidar processing for lane detection and estimation," in *ITSC*, 2009.

[9] M. Thuy and F. P. Leon, "Lane detection and tracking based on lidar data," in *Metrology and Measurement Systems*, 2010.

[10] A. Hata and D. Wolf, "Road marking detection using lidar reflective intensity data and its application to vehicle localization," in *ITSC*, 2014.

[11] A. S. Huang and S. Teller, "Probabilistic lane estimation using basis curves," in *RSS*, 2010.

[12] A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, "Finding multiple lanes in urban road networks with vision and lidar," *Autonomous Robots*, 2009.

[13] T. Wu and A. Ranganathan, "A practical system for road marking detection and recognition," in *IV*, 2012.

[14] H. Jung, J. Min, and J. Kim, "An efficient lane detection algorithm for lane departure detection," in *IV*, 2013.

[15] F. Zhang, H. Stahle, C. Chen, C. Buckl, and A. Knoll, "A lane marking extraction approach based on random finite set statistics," in *IV*, 2013.

[16] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. S. Hong, S.-H. Han, and I. S. Kweon, "Vpgnet: Vanishing point guided network for lane and road marking detection and recognition," in *ICCV*, 2017.

[17] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, "Deeplanes: End-to-end lane position estimation using deep neural networks," in *CVPR Workshop*, 2016.

[18] B. He, R. Ai, Y. Yan, and X. Lang, "Accurate and robust lane detection based on dual-view convolutional neutral network," in *IV*, 2016.

[19] A. Kheyrollahi and T. P. Breckon, "Automatic real-time road marking recognition using a feature driven approach," *Machine Vision and Applications*, vol. 23, 2012.

[20] D. Hyeon, S. Lee, S. Jung, S.-W. Kim, and S.-W. Seo, "Robust road marking detection using convex grouping method in around-view monitoring system," in *IV*, 2016.

[21] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using fully convolutional neural networks," in *IV*, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[23] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," 2017.

[24] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *NIPS*, 2015.

[25] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.