2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Uncertainty-Aware Fast Curb Detection Using Convolutional Networks in Point Clouds

Younghwa Jung, Mingu Jeon, Chan Kim, Seung-Woo Seo, and Seong-Woo Kim

Abstract— **Curb detection is an essential function of autonomous vehicles in urban areas. However, curbs are difficult to detect in complex urban environments in which many dynamic objects exist. Additionally, curbs appear in a variety of shapes and sizes. Previous studies have been based on the traditional pipeline, which consists of the extraction and aggregation of hand-crafted features that are then fed to classifiers. However, this sequential process is inefficient and designing the hand-crafted features is a complex process. Recently, this kind of process has been replaced by Deep Neural Networks (DNN), in which classifiers and features are learned from large-scale data. Very few works have exploited DNN for the curb detection problem. Most works use multi-modal sensor-based methods that combine images and accumulated 3D point clouds from LIDAR. However, these approaches require synchronization and calibration between sensors. In addition, they do not quantify the uncertainty of their predictions for autonomous system safety. In this paper, we present a two-stage DNN-based curb detection method that includes uncertainty quantification. An autoencoder-based network predicts the curbs, and then conditional neural processes rectify the predictions with uncertainty estimations. The experimental results show that our approach achieves high accuracy and recall in complex areas. We also constructed a large-scale dataset to create benchmarks consisting of approximately 5,224 scans with bird's-eye view labels collected from urban areas. To the best of our knowledge, there are no public datasets for DNN-based curb detectors. The benchmarks and datasets are publicly available at `https://github.com/YounghwaJung/curb_detection_DNN`.**

I. Introduction

Curb detection is an important function for autonomous vehicles operating in urban areas. Curbs provide road boundary information. Furthermore, detected curbs can be used to embed semantic information into raw 3D point cloud maps. LIDAR has been used as the main sensor for curb detection because it provides accurate distance measurements under adverse weather conditions. One fundamental process of curb detection using LIDAR is that of exploiting geometric properties such as gradients and height information as there are abrupt height changes between drivable roads and curbs [1]–[4].

Most state-of-the-art algorithms that use LIDAR for real-time curb detection follow a traditional sequential process [5]–[7]. The pipeline often consists of a pre-processing stage like candidate extraction, an aggregation stage for hand-crafted features, a classifying stage, and a post-processing stage such as line-fitting or line-parameter estimation. This approach has been frequently used in autonomous driving systems because it is interpretable and analytic for use in safety-critical functions. However, this sequential process is inefficient because erroneous outputs at the early stages in the pipeline affect the downstream stages. Additionally, these traditional approaches usually depend on rule-based processes, requiring heavy parameter tuning [1], [2], [5], [7]–[9]. Therefore, performance is affected by road type (e.g., curvy roads, intersection) because the optimized parameters vary according to the environment.

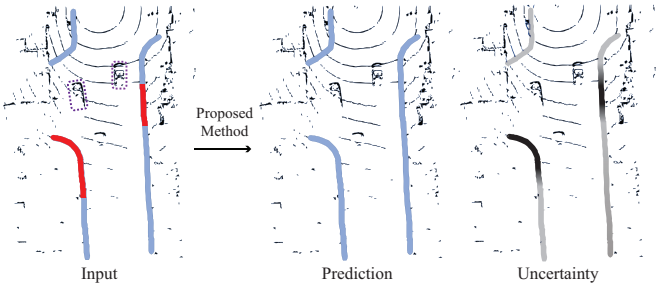

Fig. 1. Left side of the figure shows a driving environment where the purple-dotted bounding box represents the dynamic objects and the red area refers to the occluded region. The proposed method uses the encoded point clouds to estimate the location of the curbs **(light blue)** and the uncertainty of its predictions **(gray scale)** in real-time.

With advancements in Deep Neural Networks (DNN), few studies have attempted to exploit DNN to detect curbs or road boundaries with LIDAR [10], [11]. In [10], camera data and registered 3D point clouds were used as input to extract road boundaries to build semantic maps. The authors in [11] also used accumulated 3D point clouds that were registered using Visual Odometry (VO). Compared to the traditional approaches, these methods are robust to a wide range of driving environments and relieve the burden of manual parameter tuning because DNN learns target functions from data automatically. However, there is a barrier to the usage of DNN in autonomous vehicle systems because DNN offers few safety guarantees [12], [13]. Additionally, these multi-modal methods that combine images and LIDAR require time synchronization and calibration between sensors, which makes their algorithms more sensitive to sensor failure modes.

One of the challenges with the curb detection problem is predicting curbs in areas where occlusion is created by dynamic objects, as shown in Fig 1. For those undetected areas, uncertainty quantification is required to guarantee the safety of autonomous driving. However, equipping a DNN method with an uncertainty evaluation is not trivial. It is important to note that neither one of the DNN-based curb

The authors are with Seoul National University, South Korea. S. Kim is the corresponding author. E-mail: xzxzmmnn@snu.ac.kr, mingu-jeon@snu.ac.kr, chan_kim@snu.ac.kr, sseo@snu.ac.kr, snwoo@snu.ac.kr.

detection methods [10], [11] generates uncertainty estimates for their predictions.

In this study, we propose a two-stage DNN-based curb detection method that uses LIDAR. In the first stage, we detect curbs that are mostly in visible areas. For safety, we rectify the detected curbs and quantify the uncertainty of estimations by exploiting Conditional Neural Processes (CNPs) [14] in the second stage. With the LIDAR-only sensor, the proposed method avoids the synchronization and calibration problems caused by the multi-modal approaches.

The key to robust performance in deep learning-based methods is a large dataset collected under a variety of environments. However, most of the public datasets for curb detection are limited to the camera domain. The available datasets are also insufficient for deep learning methods. For example, Zhang *et al.* made a public dataset for curb detection in [5]. It contains 200 scans collected in five different scenarios. The curbs are manually labeled in each frame of the scenarios. Chen *et al.* also provided a dataset containing 566 scans collected from various urban scenes. Recent works [10] and [11] did not release their curb dataset publicly. Therefore, we built a curb detection benchmark consisting of approximately 5,200 scans with Bird's-Eye View (BEV) labels collected from urban areas and we made this dataset public.

The contributions of this paper can be summarized as follows:

- We propose a two-stage curb detection method with uncertainty quantification for safety, especially in occluded areas.
- We built a curb detection benchmark dataset, which was collected from a complex urban area. The dataset consists of approximately 5,200 scans with labels.
- We compare the proposed method to several baselines; our method shows the best performance in terms of accuracy and recall for the benchmark dataset.

The rest of this paper is structured as follows. The next section, Section II reviews related works. Section III presents the proposed algorithm in detail. Experimental results are provided in Section IV. Finally, this work is concluded in Section V.

II. Related Works

There are a large number of works on curb detection using 3D point clouds from LIDAR. We divide the LIDAR-based curb detection methods into two approaches: manual-feature-based, and deep learning-based.

A. Manual-Feature-Based Approach

Most previous works exploit a sequential process [1], [2], [5]–[8]. Hand-crafted features are extracted and then classifiers or rules are applied to detect curbs. In [5], spatial features were extracted, and four rules were applied: continuity rules (two) for the vertical and horizontal axes, an elevation rule, and an angle rule. Similarly, [2] detected curb points from ground points based on spatial features such as the gradient, normal value, and height difference. In [9], the look-ahead and the orientation are extracted as hand-crafted features. However, these approaches are inefficient because incorrect outputs at the early stages affect the stages that follow. Additionally, designing hand-crafted features requires expert knowledge. It is also difficult to find well-fitted classifiers for features defined. In practice, these kinds of properties make it difficult to apply manual-feature-based methods to self-driving platforms because the platforms must contend with a variety of environments.

B. DNN-Based Approach

Recently, a few studies have been proposed to detect curbs or road boundaries using deep neural networks [10], [11]. Liang, *et al.* [10] proposed a non-real-time-based method to generate a structured polyline corresponding to a road boundary. The extracted polylines were used to build semantic maps. They used both camera and LIDAR data and exploited the Iterative Closest Point (ICP) to make the point clouds dense. In [11], a real-time-based method was proposed to provide curb information in current driving environments. It exploited two-stage strategies in object detection tasks [15], [16] to detect the curbs in occluded areas. In the first stage, the curb detection problems were formulated with pixel-wise segmentation, and only visible curbs were detected. In the latter stage, the detected pixel-wise curbs were converted to lines by dividing the curbs into a grid. In each grid, a trained network finds the most similar predefined anchors to a line and estimates the angle and distance offset. This method also accumulated 3D point clouds by using VO algorithms with camera data.

In contrast to these approaches, the proposed method uses a single, raw 3D point cloud as input without an accumulation process. Although 3D accumulated point clouds are preferred for extracting meaningful features, registering several scans from LIDAR on heavily congested roads is more likely to produce inaccurate accumulated 3D point clouds and to increase time complexity. As in [11], we took a two-stage approach to detect curbs. However, we exploited a DNN-based Bayesian method to quantify uncertainty of curb predictions for safety.

C. Uncertainty Quantification

Bayesian probability theory has long been the mathematical tool used to estimate model uncertainty. Gaussian Processes (GPs) are a popular Bayesian method [17]. GPs define distributions over possible functions and thus, can estimate the uncertainty of predictions. The mean is itself a useful prediction, and the variance is a good way to capture uncertainty [14]. With their probabilistic properties, GPs have been extensively used in various fields [18]–[20]. However, GPs quickly become computationally intractable as the dimensionality or the data set grows. This makes it difficult for GPs to be employed in domains with large datasets. Recently, Garnelo, *et al.* [21] proposed Neural Processes (NPs) to overcome the limitations of GPs. NPs are a neural network-based method that learns an approximation

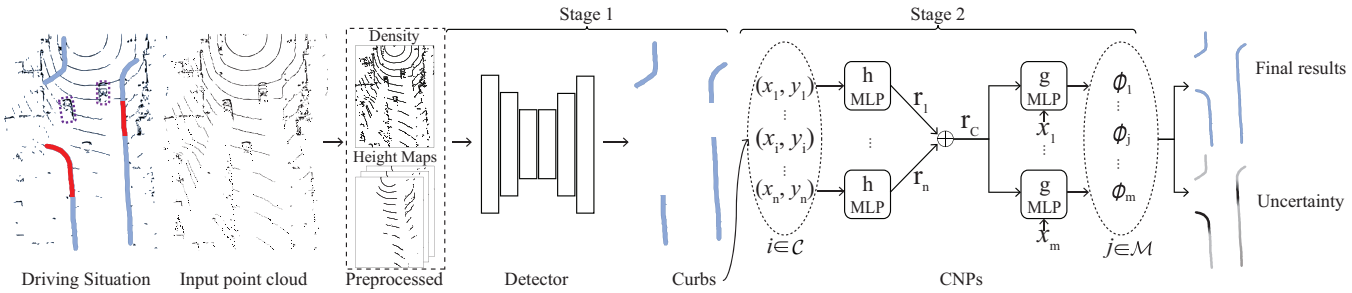

Fig. 2. Proposed curb detection framework. The red area and purple bounding boxes represent the occluded area and the dynamic objects, respectively. The point clouds are encoded by a height and a density map. In the first stage, the detector takes a pre-processed point cloud as input and estimates the curbs. Then, the detected curbs $\mathcal{C}$ are fed to the CNPs, and the probabilistic distributions for curb occupancies over $\mathcal{M}$ are generated. The final results are the corrected results for detection and the estimated uncertainty.

to stochastic processes such as GPs. NPs directly parameterize stochastic processes that map input x_i to output y_i. In particular, CNPs [14] learn the approximation of conditional distributions $p(y_T|x_T, x_C, y_C)$, where $x_C, y_C = (x_i, y_i)_{i\in C}$ and $x_T, y_T = (x_i, y_i)_{i\in T}$ are a set of context observations and targets, respectively. After the CNPs are trained, their prediction complexity is linearly related to the number of observations, and they achieve a run-time complexity of $\mathcal{O}(n + m)$ to make m predictions given n observations. With their computational efficiency and ability to quantify uncertainty, CNPs have been exploited in safety-critical applications such as trajectory and environment prediction for autonomous platforms [22]–[24]. To the best of the authors' knowledge, CNPs have not been employed in modeling the uncertainty of detection tasks with LIDAR.

III. Proposed Method

In this paper, we propose a two-stage curb detection with uncertainty quantification. In the first stage, we formulated curb detection as a pixel-wise semantic segmentation task with BEV representations of 3D point clouds. We adapted one of the representative deep architectures for semantic segmentation [25] to achieve our goal. Then, in the second stage, CNPs were exploited to rectify the detected curbs and quantify the uncertainty of the corrected detection results. The overall architecture is presented in Fig. 2.

A. BEV Representation

We exploited a compact representation by projecting 3D point clouds as a BEV. The BEV is more appropriate than a camera perspective because a flat 2D world is often assumed for path planning and for the vehicle control algorithm in autonomous driving [26]. The BEV representation of the 3D point cloud was encoded by a height and density map that was based on the encoding process in [27]. Specifically, we projected a 3D point cloud as a 2D grid map, $\mathcal{M} \in \mathbb{R}^2$, with a resolution of 0.1 m. For the height map, the maximum height of the points for each cell were used as a representative value. To extract more detailed height information, a point cloud was split into K slices. A height map was computed for each slice, thereby producing K height maps. The density map referred to the number of points in each cell. According to LIDAR resolution, the densities were normalized as $min(1.0, \frac{G+1}{L})$, where G is the number of points in the grid and L is the number of LIDAR lasers. The encoded inputs are shown in the dotted box in Fig. 2.

B. Curb Detection for Visible Curbs

We exploited the U-Net [25] as a basic architecture at the first stage. The U-Net is an encoder-decoder architecture with skip connections that are used to transfer fine-grained information from the low-level layers to the high-level layers. The U-Net-based detector takes $(1+K)$-channel tensors of size $\mathcal{M}_I \in \mathbb{R}^{H\times W\times(1+K)}$ as input and generates pixel-wise classified output $\mathcal{M}_\mathcal{C} \in \mathbb{R}^{H\times W\times 1}$, where W and H represent the spatial size. Specifically, it consists of a downsampling path and an upsampling path. The downsampling path consists of four blocks, including two convolutional layers. Each block decreases the spatial size of the feature maps by half through a maxpooling layer. After each convolutional layer, we applied a rectified linear unit layer. In the upsampling path, we upsampled the feature map through a transposed convolutional layer and concatenated the upsampled feature map with the corresponding feature map from the downsampling path. This process was repeated four times to construct the dense feature maps. Finally, the dense feature maps were used to infer the curbs. A prediction y_{pred} for each pixel was computed using a pixel-wise softmax function over the feature map. To train the network, we use the pixel-wise cross-entropy loss as the objective function [28]. The loss is summed over all the pixels in $\mathcal{M}$ as follows:

$$-\frac{1}{W\times H}\sum_{i,j}^{H,W} y_{label}^{i,j}\log(y_{pred}^{i,j}) + (1-y_{label}^{i,j})\log(1-y_{pred}^{i,j}), \tag{1}$$

where $y_{label}^{i,j}$ is a label. Although this architecture can detect visible curbs, it has limited ability to determine whether curbs exist or not in occluded areas, as shown in Section IV-E. Hence, we took a different approach to the prediction of occluded curbs.

C. Uncertainty Quantification for Curb Detection

At the second stage, we inferred the curbs in occluded areas and quantified the final detection uncertainty. Inspired

Fig. 3. (a) Curb information is extracted from the semantic map and overlaid on the satellite image as red lines. (b) Projected 3D points of curbs from semantic maps. (c) Ground truth for curb detection. (d) and (e) are the sampled training data paired with ground truth (c) for the CNPs.

by the map prediction task [24], we formulated the prediction in occluded areas as a regression problem, so that it was similar to an image completion task [29]. At the same time, the probability distributions for the final detection were generated by CNPs. In particular, we obtained a pixel-wise classified local map $\mathcal{M}_c$ from the detection module in Section III-B. Each pixel contains a 1 or a 0 to represent curb and non-curb areas, respectively. To quantify the uncertainty of the predictions, we interpreted this local map $\mathcal{M}_\mathcal{C}$ as an occupancy grid map [30]. Given only the location and occupancy of the detected curbs, we estimated the probability of occupancy over all locations of $\mathcal{M}$. Specifically, we took each pixel value $y_i \in \{0, 1\}$ as a realization of the random variable Y_i. The pixel coordinate $x_i \in \mathbb{R}^{H\times W}$ was normalized to $[0, 1]^2$. We defined the context set $(x_\mathcal{C}, y_\mathcal{C})$ and target set $(x_\mathcal{T}, y_\mathcal{T})$, where $\mathcal{C} = \{i|(x_i, y_i) \in \mathcal{M}_c, y_i = 1\}$ and $\mathcal{T} = \{i|(x_i, y_i) \in \mathcal{M}\}$. The CNPs took $(x_\mathcal{C}, y_\mathcal{C})$ as an input and generated the probabilistic distribution for the existence of curbs per pixel.

The CNPs consisted of three sub-modules, as follows:

$$\begin{aligned} r_i &= h(x_i, y_i), i \in \mathcal{C}, \\ r_C &= \bigoplus_i^n r_i, i \in \mathcal{C}, \\ \phi_k &= g(x_k, r), k \in \mathcal{T}, \end{aligned} \tag{2}$$

where h, $\bigoplus$, and g are an encoding network, an aggregation function, and a decoding network, respectively. Each context pair (x_i, y_i) passes through the encoding network and the embedding r_i is produced. Then, these embeddings are aggregated into a conditional vector r_C. The aggregation function is usually the mean operation $r_C = (r_1 + ... + r_n)/n$. However, this mean-aggregation makes it difficult for the decoder to learn which context points provide important information for a given target because the function gives the same weight to each context point. Therefore, we exploited the attention mechanism-based aggregation method proposed in [31]. Lastly, the decoding network generates a set of parameters $\phi_k = (\mu_k, \sigma_k)$ of a distribution for the occupancy of curbs over $x_\mathcal{T} \in \mathbb{R}^2$, where μ and σ are the mean and the variance, respectively. Then, the predictive distribution is $\mathcal{Q}_\theta(y_k|x_k) = p(y_k|x_k, \phi_k, \theta)$, where θ refers to learnable CNP parameters. We trained the CNP networks by minimizing the conditional negative log-likelihood,

$$\mathcal{L} = -\mathbb{E}_{|\mathcal{C}|}\Big[\sum_{k=0}^{|\mathcal{T}|} \log Q_\theta(y_k|x_k)\Big], k \in \mathcal{T}, \tag{3}$$

using the Adam optimizer [32].

TABLE I
CHARACTERISTICS OF BASELINES BASED ON DNN.

Baselines	[25]	[28]	[26]
Skip Connection	✓		
Strided Conv		✓	
Dilated Conv			✓
Max Pool	✓		
Symmetric	✓		

IV. EXPERIMENT

A. Experimental Setup

The experimental driving platform was equipped with one Velodyne VLP-32C, an OXTS RT3002, and various other sensors [33]. The dataset used to train the deep neural networks was collected using our autonomous driving vehicle. We had a semantic map that contained manually labeled information about curbs, road markings, traffic lights, etc. We extracted curb information from the semantic map as shown in Fig. 3(a). The obtained curb information consisted of 3D points. We built the ground truth from curb information in the BEV representations, as shown in Fig. 3(b)-(e). We trained the detection network on the dataset for 500 epochs using a Titan Xp GPU. The CNPs were trained on the augmented dataset for 50 epochs. The detection network and the CNPs were trained separately. The dataset used for training and testing is available online[1].

B. Generating Training Data from Semantic Maps

The curb information from the semantic map was 3D points with respect to global coordinates. We transformed the curb information from global to local coordinates using the localization results from our driving platform [33] and extracted the 3D points in the range of $[-16, 16] \times [0, 41.6]$ m, centered at the vehicle location. The extracted 3D points were then projected into the 2D plane, as shown in Fig. 3(b). However, we cannot use this kind of data to train the detector directly because the number of pixels corresponding to the curbs is extremely small compared with the total number of pixels in $\mathcal{M}$. This imbalance in the data prevents the DNN-based approach from learning, which is shown in the ablation study of Section IV-G. Therefore, we generated the interpolated pixel-wise labels by connecting the projected label points for curbs and adding a margin (e.g., 1 pixel) on both sides, as shown in Fig. 3(c). The margin considers that curbs span a vertical region between the road and sidewalk.

[1]The dataset with ground truth can be found at https://github.com/YounghwaJung/curb_detection_DNN

TABLE II
COMPARISON OF RESULTS.

Method (unit)	Precision	Recall	F-1 score
Proposed	**0.9391**	**0.9427**	**0.9408**
[25]	0.9011	0.9028	0.9019
[28]	0.8494	0.8297	0.8394
[26]	0.8834	0.7766	0.8265

TABLE III
PROCESSING TIMES FOR THE TEST DATASET.

Module	Mean (ms)	Standard Deviation (ms)
Detector	36	0.08
CNPs	52.5	0.002
Total	88.5	0.082

We used the interpolated label as the ground truth to train the curb detector.

We also generated the training data for the CNPs. We used the detection results from the detector as context and the ground truth as target. However, in the real world, occlusion cases are rather rare, which makes the occlusion situation physically non-trivial. Therefore, the dataset generated by the curb detector was highly imbalanced for training the CNPs. Inspired by image inpainting tasks [29], we augmented the dataset by modifying the ground truth. Specifically, we defined several rectangular masks representing dynamic objects such as vehicles and pedestrians, then placed them randomly in the ground truth. The number of masks was also randomly selected in the range [1, 5]. The masked ground truths for the CNPs are shown in Fig. 3(d), (e). These masked ground truths were used as contexts paired with targets in Fig. 3(c).

C. Metrics

We applied the quantitative metrics used in the KITTI-Road dataset [34] because it is based on a pixel-based evaluation in a BEV representation. We used three metrics: precision, recall, and F-1 score. The pixels having means estimated by the CNPs to be above 0.5, as described in Section III-C, were classified as curb pixels. Considering the spatial resolution of the grid map, we used a tolerance of 1 pixel, which means that the detected curbs that were located within 10 cm of ground truth were true positive.

D. Baseline Implementations

We implemented several baselines by using representative deep architectures for semantic segmentation [25], [28]. These architectures can be used for LIDAR-based detection tasks effectively [35], [36]. The Fully Convolutional Network (FCN) [28] and U-Net [25] are similar in that they have downsampling and upsampling paths. The FCN upsamples only once and exploits strided convolutional operations in the downsampling path. In contrast to the FCN, the U-Net has multiple upsampling layers, which make the U-Net symmetric, and uses max pooling in downsampling.

TABLE IV
IMPACT OF AN IMBALANCED DATASET.

Type	Precision	Recall	F-1 score
SP	0.2640	0.2461	0.2547
IP	0.9391	0.9427	0.9408

Additionally, the U-Net has skip connections that transfer coarse contextual information to the upsampling path. We also implemented a popular road detection method based on LIDAR [26]. The differences between our tasks are the spatial size and the detection target. The characteristics of each DNN-based baseline are shown in Table I.

E. Quantitative Evaluation

We evaluated the performance of the proposed method to demonstrate its effectiveness. To the best of our knowledge, there is no public dataset or benchmark for data-driven curb or road boundary detection methods. Hence, we performed a quantitative evaluation based on our dataset, and made the dataset public[1]. We compared the proposed method with the baselines in Section IV-D. To conduct a fair comparison, we adapted the baselines to comply with the evaluation metric and reduced the number of segmentation classes to one. Table II shows that the proposed curb detection method outperforms other algorithms [25], [26], [28] in terms of precision, recall, and F1-Score. Specifically, our method obtains improvements of approximately 4%, 11%, and 17% in terms of recall over the baselines. This is because the baselines cannot capture contextual information around dynamic objects because of their limited receptive field, which makes them unable to estimate curbs in occluded areas. In contrast, the refinement process with CNPs makes it possible to infer the curbs effectively in occluded areas. Similarly, we achieved better precision than that of other methods. This proves that the mean estimates of probability distributions are accurate.

The processing time for the proposed method was evaluated. We decided that 100 ms was a minimum requirement for real-time processing because the localization update cycle of our platform was 100 ms. Table III shows the mean and standard deviations of the algorithm. The overall time complexity of the CNPs is linear in relation to the number of observations. In addition, most of the operations in the detection network were based on simple convolutional layers [37]. Therefore, we achieved a real-time requirement that is essential for self-driving cars.

F. Qualitative Evaluation of Uncertainty

A quantitative evaluation of uncertainty estimation is not feasible because it requires a true probability distribution about the real world. However, the uncertainty quantification of CNPs has already been determined in several studies using a simulated experimental setup [14], [21], [31]. Therefore, we evaluated the uncertainty estimation qualitatively. We used the variance of the probability distribution for each $x_{i \in \mathcal{T}}$ as uncertainty. A high variance indicates that the

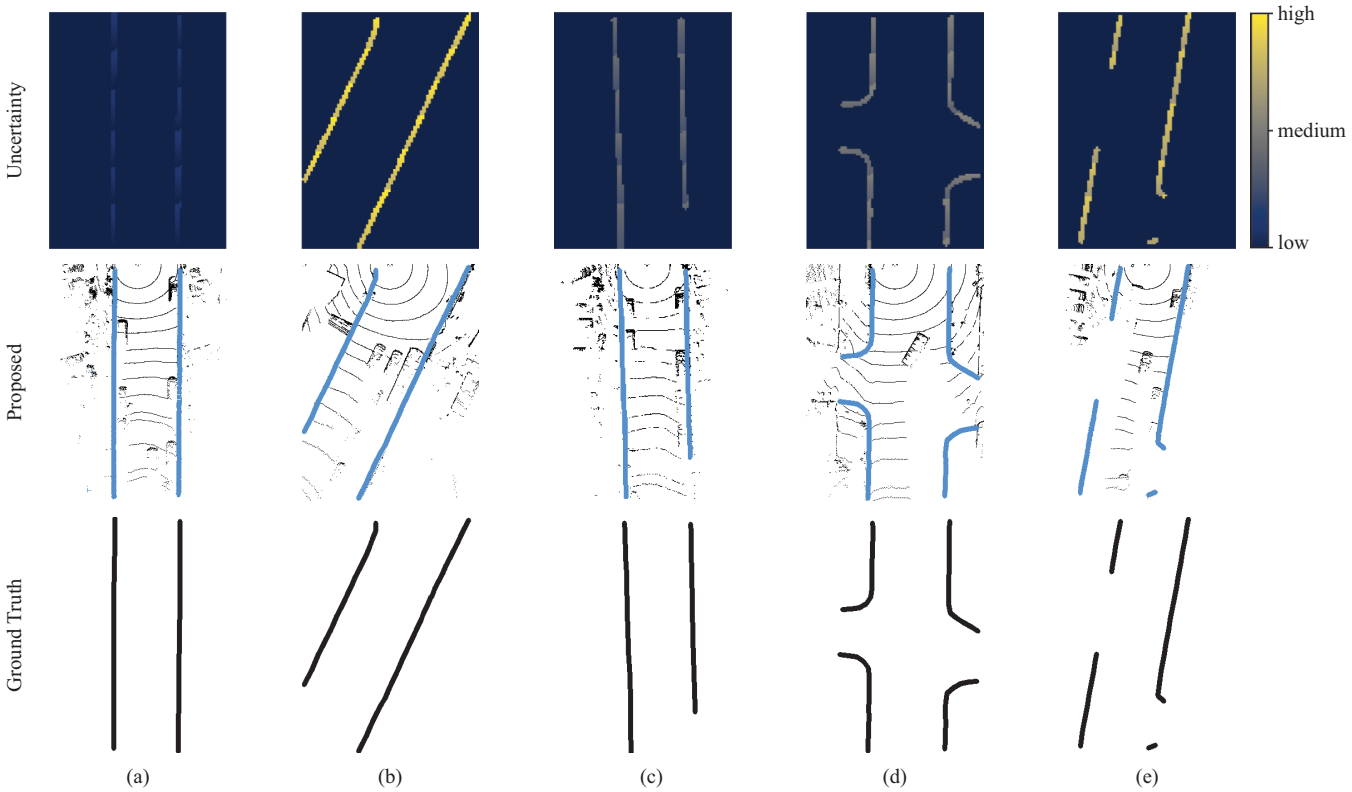

Fig. 4. Qualitative evaluation of the proposed method. **Top row:** uncertainty quantification. **Second row:** the final detection results of the proposed method. The cyan pixels represent the detected curbs. **Bottom row:** the ground truth.

sampled data points are very spread out from the mean. Generally, in a data-driven approach, a rare sample has higher model uncertainty than a sample that appears more often in the training data [38]. Hence, in our problem, curbs with an unusual shape, e.g., an intersection or a roundabout, have higher uncertainty than curbs with a common shape, e.g., straight. Curbs with a deformed shape caused by dynamic objects also have a high prediction uncertainty. The qualitative experimental results are shown in Fig. 4. Although there are dynamic objects, the uncertainty for straight curbs is lower than for other shapes as shown in Fig. 4(a), (c). This is because straight curbs are the most common in the training dataset. In the case of Fig. 4(b) and (e), the curbs are slanted from our platform's perspective, which is somewhat uncommon. Additionally, there are several dynamic objects, thus, the uncertainty for these cases is much higher than for the others. The curbs in intersections are rare cases, but there is only one dynamic object in Fig. 4(d). Hence, the uncertainty is not higher than cases (b) and (e) in Fig. 4.

G. Ablation Study

We observed the class imbalance problem from the Simply Projected (SP) pixel-wise label, as shown in Fig. 3(b). As shown in Table IV, the proposed method was not trained with the SP pixel-wise label. This is because the discontinuous series of points offered vague information that did not enhance the learning of the deep neural network. Thus, we generated the Interpolated (IP) pixel-wise labels by connecting the projected label points for curbs as shown in Fig. 3(c). With this training dataset, the proposed method was trained as shown in Table IV.

V. Conclusions

In this work, we proposed a two-stage curb detection method that exploits one single-shot point cloud without point cloud accumulation processing, which requires tedious pre-processing and a high level of precision. This is important because curb detection tasks must be accomplished in real-time on a computationally limited platform. For safety, which is essential for self-driving cars, the proposed method estimates uncertainty for the prediction by exploiting CNPs. Meanwhile, there are several datasets for LIDAR-based curb detection methods. However, the available datasets are insufficient for deep learning methods. To the best of the authors' knowledge, there are no public datasets for data-driven methods. Therefore, it is difficult to perform a quantitative analysis with other existing algorithms. To solve this problem, we built a curb detection dataset collected from complex urban areas and made it available to the public.

Acknowledgment

This work was financially supported by the National Research Foundation of Korea through the Ministry of Science and ICT under Grant 2018R1C1B5086557, by the Ministry of Culture, Sports and Tourism, and the Korea Creative Content Agency, and by the Institute of Engineering Research at Seoul National University provided research facilities for this work.

REFERENCES

[1] A. Y. Hata, F. S. Osorio, and D. F. Wolf, "Robust curb detection and vehicle localization in urban environments," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 1257–1262.

[2] G. Zhao and J. Yuan, "Curb detection and tracking using 3d-lidar scanner," in *2012 19th IEEE International Conference on Image Processing*. IEEE, 2012, pp. 437–440.

[3] G. Vosselman and L. Zhou, "Detection of curbstones in airborne laser scanning data," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, no. Part 3/W8, pp. 111–116, 2009.

[4] L. Zhou and G. Vosselman, "Mapping curbstones in airborne and mobile laser scanning data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 18, pp. 293–304, 2012.

[5] Y. Zhang, J. Wang, X. Wang, and J. M. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3d-lidar sensor," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3981–3991, 2018.

[6] S. Xu, R. Wang, and H. Zheng, "Road curb extraction from mobile lidar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 996–1009, 2016.

[7] T. Chen, B. Dai, D. Liu, J. Song, and Z. Liu, "Velodyne-based curb detection up to 50 meters away," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 241–248.

[8] B. Qin, Z. Chong, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Curb-intersection feature based monte carlo localization on urban roads," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2640–2646.

[9] W. S. Wijesoma, K. S. Kodagoda, and A. P. Balasuriya, "Road-boundary detection and tracking using ladar sensing," *IEEE Transactions on robotics and automation*, vol. 20, no. 3, pp. 456–464, 2004.

[10] J. Liang, N. Homayounfar, W.-C. Ma, S. Wang, and R. Urtasun, "Convolutional recurrent network for road boundary extraction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9512–9521.

[11] T. Suleymanov, L. Kunze, and P. Newman, "Online inference and detection of curbs in partially occluded scenes with sparse lidar," *arXiv preprint arXiv:1907.05375*, 2019.

[12] R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, R. Cipolla, and A. Weller, "Concrete problems for autonomous vehicle safety: advantages of bayesian deep learning," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 4745–4753.

[13] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," *arXiv preprint arXiv:1811.06817*, 2018.

[14] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, and S. A. Eslami, "Conditional neural processes," in *International Conference on Machine Learning*, 2018, pp. 1690–1699.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[17] C. E. Rasmussen and C. K. Williams, "Gaussian processes for machine learning," *Gaussian Processes for Machine Learning, by CE Rasmussen and CKI Williams. ISBN-13 978-0-262-18253-9*, 2006.

[18] M.-O. Shin, G.-M. Oh, S.-W. Kim, and S.-W. Seo, "Real-time and accurate segmentation of 3-d point clouds based on gaussian process regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3363–3377, 2017.

[19] S. Choi, K. Lee, and S. Oh, "Robust learning from demonstrations with mixed qualities using leveraged gaussian processes," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 564–576, 2019.

[20] A. Damianou and N. Lawrence, "Deep gaussian processes," in *Artificial Intelligence and Statistics*, 2013, pp. 207–215.

[21] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. Eslami, and Y. W. Teh, "Neural processes," *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[22] C. Dong, Y. Chen, and J. M. Dolan, "Interactive trajectory prediction for autonomous driving via recurrent meta induction neural network," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1212–1217.

[23] J. Zhu, S. Qin, W. Wang, and D. Zhao, "Probabilistic trajectory prediction for autonomous vehicles with attentive recurrent neural process," *CoRR*, vol. abs/1910.08102, 2019. [Online]. Available: http://arxiv.org/abs/1910.08102

[24] A. Elhafsi, B. Ivanovic, L. Janson, and M. Pavone, "Map-predictive motion planning in unknown environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8552–8558.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[26] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using fully convolutional neural networks," in *2017 ieee intelligent vehicles symposium (iv)*. IEEE, 2017, pp. 1019–1024.

[27] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[29] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417–424.

[30] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 116–121.

[31] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. W. Teh, "Attentive neural processes," in *International Conference on Learning Representations*, 2018.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] S.-W. Kim, G.-P. Gwon, W.-S. Hur, D. Hyeon, D.-Y. Kim, S.-H. Kim, D.-K. Kye, S.-H. Lee, S. Lee, M.-O. Shin, *et al.*, "Autonomous campus mobility services using driverless taxi," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 12, pp. 3513–3526, 2017.

[34] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 1693–1700.

[35] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1513–1518.

[36] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[37] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5353–5360.

[38] A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3153–3160, 2020.