Off-Road Drivable Area Extraction Using 3D LiDAR Data

Biao Gao*, Anran Xu*, Yancheng Pan*, Xijun Zhao†, Wen Yao†, Huijing Zhao*
*Peking University, Beijing, China
†China North Vehicle Research Institute, Beijing, China

Abstract— We propose a method for off-road drivable area extraction using 3D LiDAR data, which aims to the autonomous driving application. A specific deep learning framework is designed to deal with the ambiguous area, which is one main challenge in the off-road environment. In order to reduce the considerable demand for human-annotated data for network training, we utilize the information from vast quantities of vehicle paths and auto-generated obstacle labels. By these auto-generated annotations, the proposed network can be trained using weakly-supervised or semi-supervised methods, which can achieve better performance with less human annotations. The experiments on our data set illustrate the reasonability of our framework and the validity of our weakly- and semi-supervised methods.

I. INTRODUCTION

Recent advances in self-driving vehicles are very impressive. The drivable area extraction is one of the key technologies in this domain and is a prerequisite for safe and reliable autonomous driving[1]. At present, the more mature techniques are mainly designed for urban structured road environment[2][3], but rare researches on off-road environment. In off-road environment, there are not structured features like traffic lanes, paved road surface or guardrails. The off-road drivable area usually has ambiguous margins, various texture and complex features which brings great challenges to the extraction of the drivable area. As a result, the algorithms designed for urban environment are difficult to directly apply to the off-road environment.

The camera and LiDAR are two main sensors that provide input data for the drivable area extraction tasks. There are many camera-based methods have already been applied in the off-road environment[4].However, the color or texture features they used are not robust enough in diverse illumination and weather conditions. The lack of 3D information limits the performance and adaptability of these methods in different scenes as well[5]. On the other hand, LiDAR has been widely used in self-driving systems because of its advantage to collect 3D point clouds data directly. There are some LiDAR-based methods depending on data segmentation and rule/threshold-based methods to extract the drivable area[6]. However, these methods rely heavily on the human-designed features and presupposed thresholds, and they usually have poor scene adaptability.

In this work, we focus on the off-road drivable area extraction using 3D LiDAR data. To illustrate the main challenge of the off-road scene, we use light blue polygons to include some typical ambiguous areas in the column (b) of Figure.1. A human driver would not like to drive on these
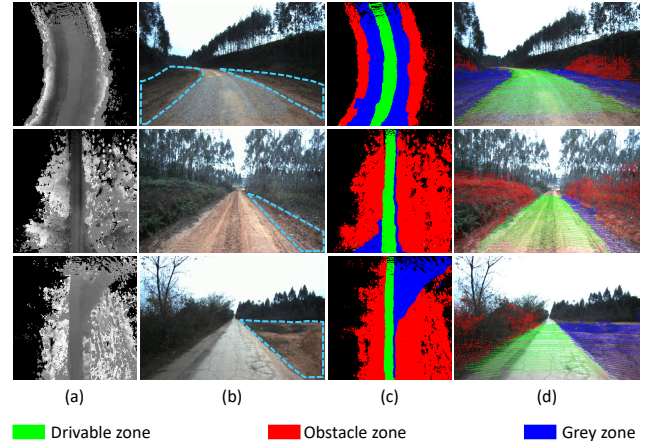

Fig. 1. The ambiguities in off-road drivable area extraction. (a) Input LiDAR data in bird's-eye view (the ego-car is in the center of the image with upward heading). (b) Image reference of input data. (c) Human annotation. (d) Projected point clouds in camera coordinate.

areas because of their higher traversability cost, but they are drivable technically in some degree. It's unreasonable to simply label these ambiguous areas as either the drivable zone or the obstacle zone, so they are called the grey zone in this paper.

We propose a deep learning method for drivable area extraction using 3D LiDAR data specific to the off-road environment. Compared with the traditional human-designed features, the proposed method can learn features of the drivable zone from the labeled data autonomously. At the same time, it is suitable for weakly-supervised and semi-supervised learning. By combining the features from the vehicle paths and auto-generated vertical obstacles, our method can significantly decrease the demand for human annotation in the neural network training. The experimental results prove the validity of our proposed method.

This paper is organized as follows: firstly, the related works are briefly introduced in Section.II. The Section.III introduces the methodology in detail. Section.IV and Section.V show some implementation details and the experimental results. Finally, we draw the conclusion in Section.VI.

II. RELATED WORKS

Camera is one of the most important sensors in the road/drivable area extraction tasks for autonomous vehicles. Some camera-based methods depend on the assumption of global road priors such as the road boundaries[7], traf-

fic lanes[8][9] or vanish points[10][11]. Some other researches do not rely on these assumptions, but view the drivable area extraction as segmentation of road and non-road regions[4][12].Despite achieving good performance, camera-based methods are easily affected by the changing of illumination. The LiDAR-based methods can fill up this weakness and the high precision 3D information can be conveniently used to extract the road boundaries[6][13][14] or fit the road plane[15][16]. For the different characters of the two sensors, LiDAR-camera fusion becomes a natural solution. For example, Dahlkamp[17] identify a nearby drivable area by LiDAR and use it to train an image-based classifier for far-range drivable area detection.

The existing drivable area extraction methods are mostly designed for the urban environment, but the problem in off-road environment is quite different. One fundamental problem is the ambiguous definition of the drivable area, as shown in Figure.1. Many researches proposed similar concepts for the off-road scenes from their own perspectives, such as traversability analysis[18] and drivable corridors[19]. Despite some methods have already been implemented to the off-road environment[11][17][19], they still have some limits. These methods usually focus on the mechanically drivable area but seldom distinguish whether these areas are likely to be chosen by a human driver, which makes great sense for autonomous vehicles.

Recently, many deep learning methods achieve impressive results on the related task[20][21][22][23][24]. Compared to the traditional methods, deep networks can learn high level semantic features from the data directly, which usually performs better than human-designed features. However, the deep learning methods usually rely on large quantities of human-annotated data set like KITTI[25] and Cityscapes[26]. For off-road environment, there is few widely-used data set for the drivable area extraction due to the ambiguous problem definition. In order to reduce the demand for human annotation, a few researches use a simulator to access endless data for training[23]. And other researches focus on weakly-supervised[20] or semi-supervised methods[18], trying to utilize the auto-generated weakly labels, which can be easily accessed, as substitutions.

This work focuses on the drivable area extraction in off-road environment. We propose a LiDAR-based deep learning framework specific to the ambiguities in this task. To reduce the demand for human-annotated data set, we also propose weakly-supervised and semi-supervised methods to learning features from auto-generated labels.

III. METHODOLOGY

A. Problem Definition

We aggregate a few frames' point clouds to get a dense bird's-eye view height map $X = \{x_{j,k}\}_{0 \le j < H, 0 \le k < W}$, which is used as our input data format. The input height map is in the size of $H \times W$ and each pixel $x_{j,k}$ means the physical height of pixel (j,k). The car is in the center of X with upward heading. The examples of input can be seen in Figure.1(a).

Different from the well-defined road borders in structured urban environment, the main peculiarity in off-road environment is the ambiguous area beside the road margin, which is called grey zone. In order to distinguish it with others, we let $LabelSet = \{unknown, drivable\ zone, obstacle\ zone, grey\ zone\}$, and we use $G = \{g_{j,k}\}_{0 \le j < H, 0 \le k < W}$ to denote human annotated ground truth, where $g_{j,k} \in LabelSet$.

The origin output of our proposed framework is a cost map $C = \{c_{j,k}\}_{0 \le j < H, 0 \le k < W}$, where each $c_{j,k} \in [0,1]$ evaluates the traversability cost of pixel (j,k). Our proposed framework learns a mapping from input X to cost map C.

$$f_\theta^* : x_{j,k} \to c_{j,k} \in [0,1] \tag{1}$$

For the convenience of comparison with human-annotated ground truth G and other baseline methods, we use d_α to remap cost map C to label $Y = y_{j,k} \in LabelSet$.

$$d_\alpha : c_{j,k} \to y_{j,k} \in LabelSet \tag{2}$$

Therefore, the problem of this work can be formulated as learning a multi-class classifier $f_\theta = d_\alpha(f_\theta^*)$ that maps input $x_{j,k}$ to a label $y_{j,k} \in LabelSet$.

B. Network Architecture

Due to the ambiguity of the grey zone, we hold the view that classifying it as an independent label from the others is not reasonable enough. In some cases, the grey zone is drivable technically but not human-desired, which is very close or even overlapped with the drivable zone in feature space. In other cases, the grey zone may have higher traversability cost than common drivable zone, which is closer or even overlapped with the obstacle zone in feature space. As a result, viewing the ambiguous grey zone as an independent label in training process will cause confusion for the deep learning model, and the experimental results in Section.V give the evidence of this viewpoint.

Therefore, we assume that the samples of grey zone are distributed between the drivable samples and the obstacle samples in feature space. **The key idea of our proposed method is learning two classification surfaces to separate the grey zone samples. One classification surface is used**
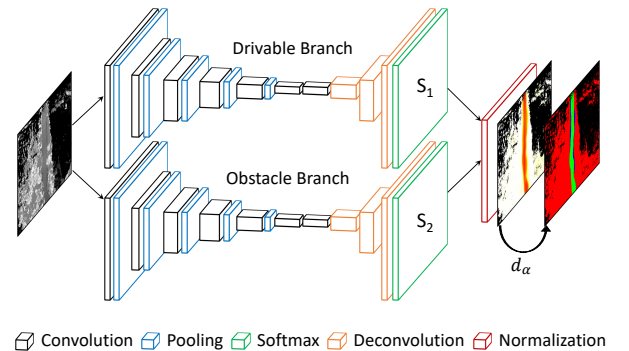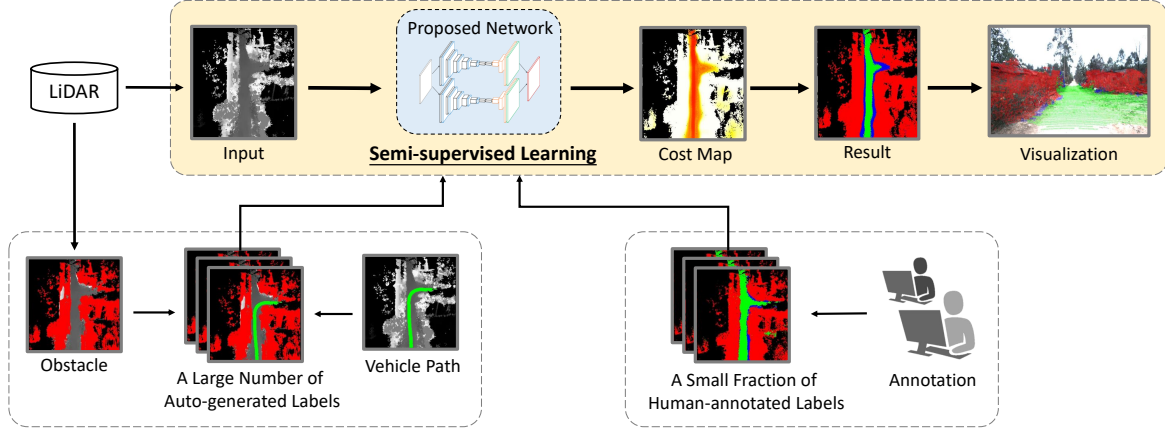

Fig. 2. The proposed network

Fig. 3. Overview of the proposed off-road drivable area extraction framework

to separate the drivable zone samples from others. The other one is used to separate the obstacle zone samples from others. As a result, the grey zone samples distribute between the two classification surfaces, and we can evaluate the traversability cost of a sample by its feature distance to the two surfaces.

As shown in Figure.2, the proposed network has two branches in order to learn the two classification surfaces mentioned above. Both of them are designed according to the common VGG-based fully convolutional network. The difference is that the last layer does not output the discrete labels but the probabilistic predictions from the last softmax layer. We denote them as S_1 and S_2, which are in the range of $[0, 1]$.

Each branch is trained end-to-end guided by the following cross entropy loss function:

$$L^{br}(X; \Theta^{br}) = -\sum_i y_i^{br} \log P(y_i^{br}|\Theta^{br}), \ br \in \{dri, obs\} \tag{3}$$

where $br \in \{dri, obs\}$ is the name of network's branch. $P(y_i^{br}|\Theta^{br})$ is the probability that pixel i is predicted as label y_i^{br} with the network parameters Θ^{br}. We use dri, obs and gre to represent $drivable, obstacle$ and $grey\ zone$.

When training the network, different label y_i^{br} is used in the two branches.

$$y_i^{br} = \begin{cases} \Psi(\vec{br}), & if\ g_i = gre \\ \vec{g_i}, & if\ g_i \in \{dri, obs\} \end{cases} \tag{4}$$

where $\vec{br}$ is the one-hot vector of label $br \in \{dri, obs\}$. $\Psi(\vec{dri}) = \vec{obs}$ and $\Psi(\vec{obs}) = \vec{dri}$. Concretely, when training the drivable branch, we replace all the pixels satisfied $g_i = gre$ with label obs as y_i^{dri}. When training the obstacle branch, we replace all the pixels satisfied $g_i = gre$ with label dir as y_i^{obs}.

We use the following regulation to calculate the traversability cost map C and the discrete label:

$$C = \begin{cases} S_1, & if\ S_1 > \alpha_1 \\ 1 - S_2, & if\ S_2 > \alpha_2 \\ \frac{1 - S_2}{1 - S_1 + 1 - S_2}, & otherwise \end{cases} \tag{5}$$

where the pixels satisfied the first condition will be labeled as $y_{j,k} = dri$, the second is $y_{j,k} = obs$ and the last is $y_{j,k} = gre$.

C. Weakly- and Semi-supervised Learning

In order to reduce the demand for the high-cost human-annotated data, we propose a semi-supervised learning method shown in Figure.3. For our weakly-supervised method, there will not having human-annotated labels using for training. And for our semi-supervised method, this framework can combine a large number of auto-generated labels (see Section.IV-A for more details) and only a small fraction of human-annotated labels to train the network. Except the numerous auto-generated labels, this framework is almost the same as the fully-supervised one. Our proposed network receives the LiDAR-based height maps as the inputs and outputs the traversability cost map for each pixel. For the convenience of evaluation, the cost map will be discreted to the 3-class result and combined with the image for visualization.

For human-annotated data X_h, we use the loss function Equation (3) for training. For data X_w with only auto-generated weakly labels, we define the loss L_{semi}^{br} in branch br as below:

$$L_{semi}^{br}(X_w; \Theta^{br}) = -\lambda \sum_i \widetilde{y_i^{br}} \log P(\widetilde{y_i^{br}}|\Theta^{br}) \tag{6}$$

where λ is a regularization weight. $\widetilde{y_i^{br}}$ is used for weakly- and semi-supervised training, which has a similar definition as y_i^{br} except for the pixels without auto-generated labels $\widetilde{g_i}$. unk means the unknown zone. $g_i \neq unk$ means the pixels without LiDAR observation, so they are labeled as unknown.

TABLE I

EVALUATION MEASURES

Drivable Zone		Obstacle Zone	
Definition	Explanation	Definition	Explanation
$Q_1 = TP(G_{dri})/\|Y_{dri}\|$	$TP(G_{dri}) = \|G_{dri} \cap Y_{dri}\|$	$Q_1 = TP(G_{obs})/\|Y_{obs}\|$	$TP(G_{obs}) = \|G_{obs} \cap Y_{obs}\|$
$Q_2 = TP(G_{dri})/\|G_{dri}\|$	$TP(G_{dri}) = \|G_{dri} \cap Y_{dri}\|$	$Q_2 = TP(G_{obs})/\|G_{obs}\|$	$TP(G_{obs}) = \|G_{obs} \cap Y_{obs}\|$
$Q_3 = TP(VP_{dri})/\|VP_{dri}\|$	$TP(VP_{dri}) = \|VP_{dri} \cap Y_{dri}\|$	/	/
$F_1 = 2Q_1Q_2/(Q_1 + Q_2)$	F_1 measure	$F_1 = 2Q_1Q_2/(Q_1 + Q_2)$	F_1 measure

dri: Drivable zone   **obs**: Obstacle zone   **G**: Ground truth   **Y**: Prediction   **VP**: Vehicle path $\|\mathbf{X}\|$: Pixel number in X

$$\widetilde{y_i^{br}} = \begin{cases} \Psi(\vec{br}), & if \ \widetilde{g_i} = unk \ \& \ g_i \neq unk \\ y_i^{br}, & otherwise \end{cases} \quad (7)$$

When training the network with human-annotated labels and auto-generated labels simultaneously, combine the Equation (3) and Equation (6) together.

$$L^{br}(X_h, X_w; \Theta^{br}) = L^{br}(X_h; \Theta^{br}) + L_{semi}^{br}(X_w; \Theta^{br}) \quad (8)$$

In each training batch, the human-annotated and auto-generated data will be randomly fed to the model, and the mean loss of them will be calculated for back propagation. If only use auto-generated label for training (weakly-supervised learning), **Equation (6) be the only loss function.**

IV. IMPLEMENTATION DETAILS

A. Automatic Labeling

As illustrated in Figure.3, we make use of amounts of recorded data from the data collection car to achieve the auto-generated labels. We follow a rules-based region grow method described in Algorithm.1 to generate the vertical obstacles from LiDAR data as the weakly obstacle zone labels. Only need to set a loose threshold for region grow, we can get a relatively strict vertical obstacle area.

Besides, we assume the vehicle path chosen by the human driver must belong to the drivable zone. So we project the data collection car's GPS trajectories with the same width as the car to the input height map, and they are labeled as the weakly drivable zone.

B. Training Setup

The training process and experiments are conducted on a NVIDIA TitanX GPU. The network is trained with Adam optimizer with the learning rate 1e-4 and the batch size of 16. Data augmentation is applied mainly for image rotation, because the vehicle seldom makes a turn. In semi-supervised learning loss, the regularization weight λ is usually less than 1, such as 0.4. When converting the cost map into discrete labels, we set the thresholds α_1 and α_2 to 0.5 for an SUV. Actually, they can be changed based on the through capacity of a vehicle.

Algorithm 1 Region Grow

Input: Height map X, Height threshold T_h, Angle threshold T_a, Initial road height threshold T_r
Output: Drivable region set S_d, Obstacle region set S_o
1: Initialize: waiting list $Q = \emptyset, S_d = \emptyset, S_o = \emptyset$
2: Delta height function: ΔH
3: Delta angle function: ΔA
4: **for all** x_i **in** X **do**
5:    **if** $x_i \in [-T_r, T_r]$ **then**
6:       $Q \leftarrow Q \cup x_i, S_d \leftarrow S_d \cup x_i$
7:    **end if**
8: **end for**
9: **while** $Q \neq \emptyset$ **do**
10:    **for all** $x_j \in NeighbourSet(x_i)$ **do**
11:       **if** $\Delta H(x_i, x_j) < T_h$ **and** $\Delta A(x_i, x_j) < T_a$ **then**
12:          $Q \leftarrow Q \cup x_j, S_d \leftarrow S_d \cup x_j$
13:       **else**
14:          $S_o \leftarrow S_o \cup x_j$
15:       **end if**
16:    **end for**
17:    $Q \leftarrow Q - x_i$
18: **end while**

C. Evaluation Measures

In order to evaluate the quantitative performance of different algorithms in the off-road environment, we design some evaluation measures in Table.I.

Due to the ambiguous definition of the grey zone, we do not evaluate the performance on the grey zone samples directly, but only the drivable zone and obstacle zone samples.

1) Precision: We define Q_1 to evaluate the precision performance. For the drivable zone, $Q_1 = TP(G_{dri})/\|Y_{dri}\|$. $\|Y_{dri}\|$ means the number of pixels predicted as the drivable zone. Q_1 measures the percentage of extracted drivable pixels that are actually the drivable zone in the ground truth. For the obstacle zone, Q_1 measures the percentage of extracted obstacle pixels that belong to the obstacle zone in human annotations.

2) Recall: We define Q_2 to evaluate the recall. For the drivable zone, $Q_2 = TP(G_{dri})/\|G_{dri}\|$, where $\|G_{dri}\|$ means the pixel number of the drivable zone in the ground truth. Q_2 measures the percentage of the ground truth driv-

able zone extracted by the proposed method. For the obstacle zone, Q_2 is defined in the similar fashion.

3) Accuracy: Q_3 is defined only for the drivable zone, which is in order to measure the percentage of the vehicle path extracted as the drivable zone. We believe that the vehicle paths chosen by the human driver must be the area with low traversability cost relatively. Therefore, we design Q_3 to encourage extracting the pixels of vehicle path as the drivable zone.

4) F_1 Measure: The last but not least, the F_1 measure is a widely-used indicator which considers both the precision and the recall. The F_1 measure is the harmonic average of the precision (Q_1) and recall (Q_2).

In this work, there are some methods tend to extract the drivable zone with narrow-width, which is similar with a vehicle path. This will lead to a very high performance of precision (Q_1) but lower recall (Q_2). Besides, there are some other methods tend to extract wider drivable zone, which will lead to higher recall (Q_2) but lower precision (Q_1). In these cases, F_1 measure will be considered as the most important indicator to evaluate the methods performance.

V. EXPERIMENTAL RESULTS

A. Data Set

We build a typical off-road data set by our data collection vehicle, which is equipped by a Velodyne HDL-64 LiDAR, a front-view monocular camera and a GPS/IMU system. To get the input data X, We project point clouds captured by the LiDAR into birds-eye view. The position and posture information captured by the GPS/IMU system are required during the projection process. Besides, we use the GPS/IMU system to record the vehicle trajectory for the automatic labeling of vehicle path. We emphasize that the camera data are only used for visualization.

During the process of data collection, the vehicle is driven by a human driver and all kinds of data are time-synchronized. The input height map X is in the size of 300×300 with 0.2 meters pixel size. The height value of each pixel is linear projected to an integer in $[0, 255]$.

The whole data set contains 1961 frames data. It's approximately about 785 meters driving distance. We choose 1176 frames for model training, 294 frames for validation and 491 frames for test.

B. Proposed Method Results

In order to evaluate our proposed method performance, we firstly compare it with a fully-supervised method to prove the reasonability of our model design for the grey zone. **Another advantage of our model is that it's also suitable for weakly- and semi-supervised learning.** So we then compare our weakly- and semi-supervised results with other baselines.

Figure.4 and Figure.5 show the qualitative test results of different methods in two typical scenes: a cross road scene and a straight road scene. It's necessary to mention that the cost maps of other baseline methods are directly remapped from their output labels, which only have 3 discrete values.

We use the evaluation measures described in Section.IV-C to compare the quantitative performance of different methods, which is shown in Table.II.

1) Fully-supervised Results: Fully-supervised results use the human-annotated ground truth for model training. The baseline method '3-class FCN' is a fully convolutional network with the same depth as either branch in our proposed network. It treats the problem as a common 3 class classification. Our proposed fully-supervised method achieves better performance than '3-class FCN' in most evaluation measures.

We can see the first three columns in Figure.4 for a more concrete example. Our fully-supervised method is robuster than the common 3-class classification method in the complex cross road scene. The baseline '3-class FCN' misclassifies the left side cross road as the obstacle zone and our method successfully extract the whole structure of this cross road.

2) Weakly-supervised Results: The weakly-supervised method means all data for model training are auto-generated weakly labels.

The qualitative visualization results are shown in the last four columns of Figure.4 and Figure.5. From the visualization results, it's easy to find that the 'RG-FCN' method tends to extract a wider drivable zone than the ground truth. The rule-based method can not distinguish the drivable zone and the grey zone with a few of thresholds, so the 'RG-FCN' tends to treat the drivable zone and the grey zone which is drivable technically as the same. The 'Oxford path proposal' results are almost to the contrary, which have very narrow drivable zone. This method can extract the drivable zone

TABLE II

QUANTITATIVE EVALUATION OF DIFFERENT METHODS

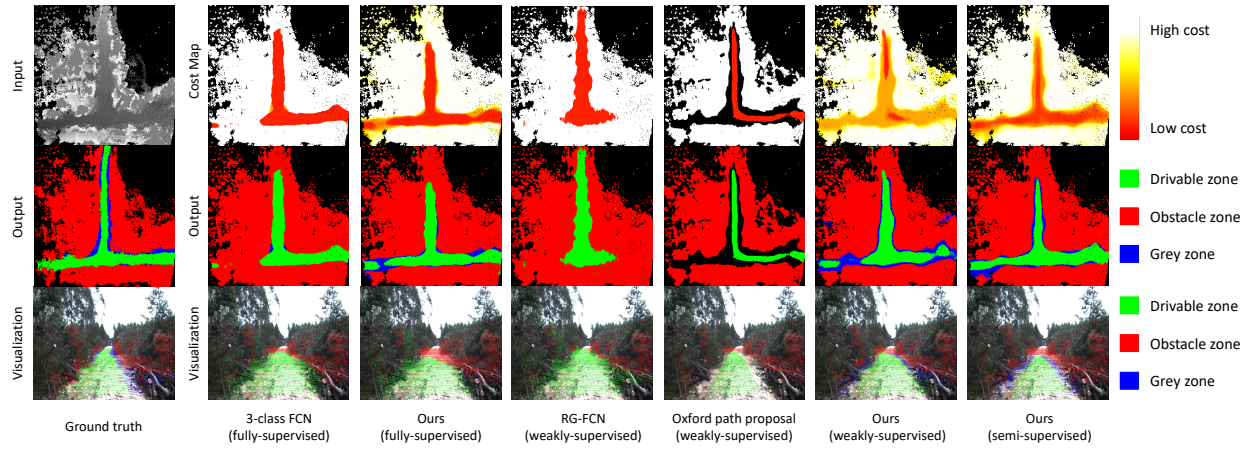|	Drivable zone				Obstacle zone		
	Q_1 (PRE)	Q_2 (REC)	Q_3 (ACC)	F_1	Q_1 (PRE)	Q_2 (REC)	F_1
3-class FCN (fully-sup.)	74.93	82.99	**98.92**	78.75	94.36	**98.44**	96.36
Ours (fully-sup.)	**76.01**	**86.72**	98.09	**81.01**	**96.20**	96.75	**96.47**
RG-FCN (weakly-sup.)	59.78	79.15	93.16	68.11	94.46	95.38	94.92
Oxford PP (weakly-sup.)	**97.00**	47.38	83.71	63.66	**98.40**	89.84	93.93
Ours (weakly-sup.)	72.38	78.83	95.21	75.47	96.31	94.84	95.57
Ours (semi-sup.)	81.73	**81.73**	**96.24**	**81.73**	95.60	**97.38**	**96.49**

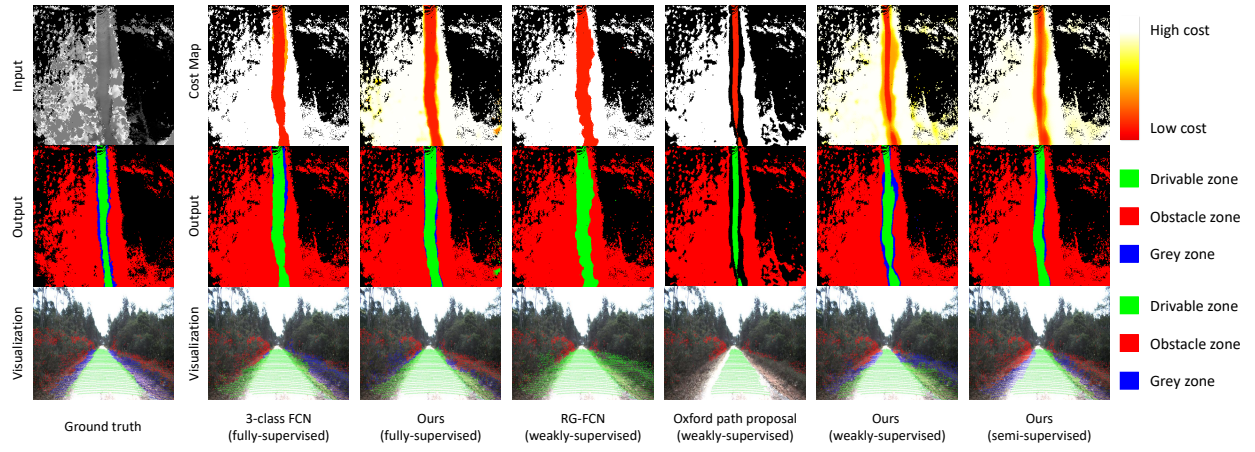Fig. 4. Qualitative results at cross road scene.

Fig. 5. Qualitative results at straight road scene.

similar to the vehicle path, but the fundamental defect is too many pixels between this narrow drivable zone and the obstacle zone are labeled as unknown. A large percentage of these pixels are actually drivable and an accurate prediction of them makes a great sense for autonomous vehicles. Compared with these two baseline methods, our weakly-supervised method has obviously better performances in extraction of the drivable area. In addition, it's also as robust as the fully-supervised method in the cross road scene.

The quantitative analysis shows that our weakly-supervised method can extract the drivable area more accurately than others. Despite one baseline method has a higher precision Q_1 and the other one has a little higher recall Q_2, they both get very poor performance of the other evaluation measure. In other words, our method is the most balanced one, so we can get the F_1 measure that 7.4% higher than the 'RG-FCN' and 11.8% higher than the 'Oxford path proposal' method.

3) Semi-supervised Results: The semi-supervised methods means a proportion of human-annotated and auto-generated data are used for training at the same time.

By the way, no matter how many human-annotated labels are contained, we will use all weakly labels for training, cause for their quite low generating cost. We list the semi-supervised result using half human-annotated labels for training as a representative in Table.II. The F_1 measure of our semi-supervised method (81.73%) achieves an impressive improvement compared to other baseline methods and even higher than the fully-supervised result. It achieves better evaluations than other methods in all measures except the precision Q_1, the explanation of this is similar with the weakly-supervised ones.

In order to explore how the ratio of human-annotated labels influence our model performance. We compare the semi-supervised methods using different ratio's human-annotated labels, base on the key indicator F_1 measure. The detailed performance on the test set can be seen in Figure.6. We split the test set into 10 batches and evaluate on them respectively. The quantitative results is shown in Table.III. The percentage in the front of 'semi-sup' means the ratio of human-annotated labels used for training. The F_1 measure of the 50% semi-supervised version is higher than the fully-

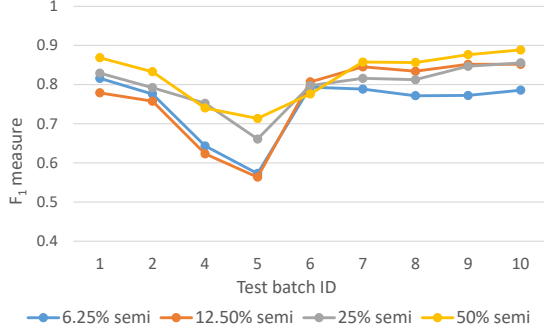| -	6.25% semi-sup.	12.5% semi-sup.	25% semi-sup.	50% semi-sup.	Fully-sup.
F_1 measure	73.86	75.04	78.96	**81.73**	81.01

Fig. 6. Detailed comparison of semi-supervised methods on the test set.

supervised method, and the 25% semi-supervised version achieves higher performance than the '3-class FCN' baseline with only a quarter of human-annotated labels. It shows that our proposed semi-supervised method can significantly reduce the demand for high-cost human annotations.

VI. CONCLUSION

In this paper, we propose a deep learning framework for off-road drivable area extraction. The proposed network structure is specifically designed for the ambiguous grey zone in the off-road environment. We also propose an automatic labeling method which generates quantities of weakly labels from the data collection vehicle's driving. It can significantly reduce the demand for human-annotated data for the weakly- and semi-supervised network training. Crucially, it's demonstrated that the proposed semi-supervised method can achieve better performance than the fully-supervised one with even less human-annotated labels. In this work, the camera images are only used for visualization, but they actually include many useful features for the drivable area extraction, such as the colors and texture. We plan to fuse the camera information into our framework and enhance the robustness of far-field drivable area extraction in the off-road environment.

REFERENCES

[1] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent progress in road and lane detection: A survey. *Machine Vision and Applications*, 25(3):727–745, 2014.

[2] Yinghua He, Hong Wang, and Bo Zhang. Color-Based Road Detection in Urban Traffic Scenes. 5(4):309–318, 2004.

[3] José M. Álvarez and Antonio M. opez. Road detection based on illuminant invariance. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):184–193, 2011.

[4] Jilin Mei, Yufeng Yu, Huijing Zhao, and Hongbin Zha. Scene-Adaptive Off-Road Detection Using a Monocular Camera. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):242–253, 2018.

[5] Liang Xiao, Bin Dai, Daxue Liu, Tingbo Hu, and Tao Wu. CRF based road detection with multi-sensor fusion. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2015-August(Iv):192–198, 2015.

[6] Wende Zhang. LIDAR-based road and road-edge detection. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 845–848, 2010.

[7] Yuan Yuan, Zhiyu Jiang, and Qi Wang. Video-based road detection via online structural learning. *Neurocomputing*, 168:336–347, 2015.

[8] Mohamed Aly. Real time Detection of Lane Markers in Urban Streets. pages 7–12, 2014.

[9] ZuWhan Kim. Robust Lane Detection and Tracking in Challenging Scenarios. *Ieee Transactions on Intelligent Transportation Systems*, 9(1):16–26, 2008.

[10] Jose M. Álvarez, Antonio M. López, Theo Gevers, and Felipe Lumbreras. Combining priors, appearance, and context for road detection. *IEEE Transactions on Intelligent Transportation Systems*, 15(3):1168–1178, 2014.

[11] Jean-yves Audibert, Jean Ponce, and Ponts Paristech. General Road Detection From a Single Image. *IEEE Transactions on Image Processing*, 19(8):2211–2220, 2010.

[12] Shengyan Zhou and Karl Iagnemma. Self-supervised learning method for unstructured road detection using fuzzy support vector machines. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pages 1183–1189, 2010.

[13] W. S. Wijesoma, K. R.S. Kodagoda, and Arjuna P. Balasuriya. Road-boundary detection and tracking using ladar sensing. *IEEE Transactions on Robotics and Automation*, 20(3):456–464, 2004.

[14] Yihuan Zhang, Jun Wang, Xiaonian Wang, and John M. Dolan. Road-Segmentation-Based Curb Detection Method for Self-Driving via a 3D-LiDAR Sensor. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2018.

[15] Alireza Asvadi, Cristiano Premebida, Paulo Peixoto, and Urbano Nunes. 3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. *Robotics and Autonomous Systems*, 83:299–311, 2016.

[16] Xiao Hu, F. Sergio A. Rodriguez, and Alexander Gepperth. A multimodal system for road detection and segmentation. *IEEE Intelligent Vehicles Symposium, Proceedings*, (Iv):1365–1370, 2014.

[17] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised Monocular Road Detection in Desert Terrain. *Robotics: Science and Systems II*, 2006.

[18] Benjamin Suger, Bastian Steder, and Wolfram Burgard. Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3D-lidar data. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June):3941–3946, 2015.

[19] Ara V. Nefian and Gary R. Bradski. Detection of drivable corridors for off-road autonomous navigation. *Proceedings - International Conference on Image Processing, ICIP*, (Figure 1):3025–3028, 2006.

[20] Dan Barnes, Will Maddern, and Ingmar Posner. Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 203–210. IEEE, 2017.

[21] Xiaofeng Han, Jianfeng Lu, Chunxia Zhao, and Hongdong Li. Fully Convolutional Neural Networks for Road Detection with Multiple Cues Integration. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4608–4613, 2018.

[22] Gabriel L. Oliveira, Wolfram Burgard, and Thomas Brox. Efficient deep models for monocular road segmentation. *IEEE International Conference on Intelligent Robots and Systems*, 2016-November:4885–4891, 2016.

[23] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730, 2015.

[24] Mauro Bellone, Giulio Reina, Luca Caltagirone, and Mattias Wahde. Learning Traversability from Point Clouds in Challenging Scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):296–305, 2018.

[25] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[26] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.