Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

GeeNet: robust and fast point cloud completion for ground elevation estimation towards autonomous vehicles*

Liwen LIU†1, Weidong YANG[1,2], Ben FEI†‡1

[1]*Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai 200433, China*

[2]*Zhuhai Fudan Innovation Institute, Zhuhai 519000, China*

†E-mail: 21210240022@m.fudan.edu.cn; bfei21@m.fudan.edu.cn

Received May 31, 2023; Revision accepted July 24, 2023; Crosschecked June 25, 2024

Abstract: Ground elevation estimation is vital for numerous applications in autonomous vehicles and intelligent robotics including three-dimensional object detection, navigable space detection, point cloud matching for localization, and registration for mapping. However, most works regard the ground as a plane without height information, which causes inaccurate manipulation in these applications. In this work, we propose GeeNet, a novel end-to-end, lightweight method that completes the ground in nearly real time and simultaneously estimates the ground elevation in a grid-based representation. GeeNet leverages the mixing of two- and three-dimensional convolutions to preserve a lightweight architecture to regress ground elevation information for each cell of the grid. For the first time, GeeNet has fulfilled ground elevation estimation from semantic scene completion. We use the SemanticKITTI and SemanticPOSS datasets to validate the proposed GeeNet, demonstrating the qualitative and quantitative performances of GeeNet on ground elevation estimation and semantic scene completion of the point cloud. Moreover, the cross-dataset generalization capability of GeeNet is experimentally proven. GeeNet achieves state-of-the-art performance in terms of point cloud completion and ground elevation estimation, with a runtime of 0.88 ms.

Key words: Point cloud completion; Ground elevation estimation; Real-time; Autonomous vehicles
https://doi.org/10.1631/FITEE.2300388 **CLC number:** TP391

1 Introduction

The progress in fully autonomous driving and intelligent robots requires reliable perception of the surrounding environment. In recent years, comprehensive three-dimensional (3D) perception of scenes has been a challenging but crucial research topic (Leonard et al., 2008). To achieve this, 3D sensors, including LiDAR and RGB-D, are widely used in autonomous vehicles and robots; they can capture 3D point clouds of the environment with high resolution and remain unaffected by varying illumination.

Due to the representations of 3D point clouds for shape features and precise position, point cloud data generated by 3D sensors such as LiDAR can be leveraged for numerous applications (Chen YD et al., 2016; Liu Y, 2016; Liu BS et al., 2019), such as 3D object detection, navigation, and path planning. In autonomous vehicle navigation, accurate and effective detection of navigable space undoubtedly plays a crucial role (Rummelhard et al., 2015). Traditionally, in typical steps that precede the navigable space, the ground is assumed as a plane without height information. The flat-ground assumption in point clouds results in systematic errors or increased computation (Rummelhard et al., 2017).

The target of ground elevation estimation is to predict the height of the ground. It is a challenging task because of the sparse nature of the point clouds. Moreover, as shown in Fig. 1, some grid

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. U2033209)
ORCID: Liwen LIU, https://orcid.org/0000-0003-1867-3046; Ben FEI, https://orcid.org/0000-0002-3219-9996

cells are occupied by temporary obstructions (such as vehicles or pedestrians), some grid cells lack points due to occlusion, and uneven ground surfaces are other challenges in estimating the ground height of the grid cells. Correctly estimating ground elevation can bring many benefits. For example, knowing the height of each LiDAR point relative to the ground can benefit many LiDAR-based 3D object detection, road detection, and lane detection instances (Zhou and Tuzel, 2018; Lang et al., 2019).

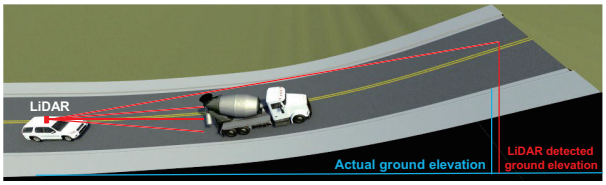

Fig. 1 Definition of the actual and detected ground elevation from LiDAR on the car. Due to ground occlusions, LiDAR-detected ground elevation is always different from the actual ground elevation

Several early works focused on projecting 3D points as a 2.5-dimensional (2.5D) grid (Thrun et al., 2006) or using a two-dimensional (2D) line extraction based fast algorithm (Himmelsbach et al., 2010). Recent methods tend to be carried out with the help of point cloud segmentation (Ren et al., 2022). For example, GndNet (Paigwar et al., 2020) was proposed as an approach for estimating the ground-level height and the height of the ground plane and segmenting the ground points at the same time. This method overcomes the difficulties of public datasets without annotated ground elevation maps. First, GndNet can operate directly on sparse 3D points and is trainable in an end-to-end manner. Second, GndNet learns appearance in real time, analyzes scenes, and estimates ground elevation.

However, these segmentation-based methods are still constrained due to the sparse nature of point clouds. Moreover, the runtime of GndNet is 18.2 ms, which is inadequate for real-time applications. To solve these problems, some methods use existing semantic scene completion. For instance, LMSCNet (Roldão et al., 2020), SSCNet (Song et al., 2017), and JS3C-Net (Yan et al., 2021) were devised for semantic scene completion in a grid-based representation. Nevertheless, these methods were designed for completion of multiple classes, which leads to complicated network architectures and high computational cost.

To avoid these shortcomings, several issues must be addressed. First, how can we effectively improve complex network architectures and reduce high computing costs? Second, as learning-based approaches require a large amount of annotated data, how can we train the model data without a public dataset for annotated ground elevation maps?

To address the first issue, we design a method that can simultaneously complete the ground height map and estimate the ground height map in real time, where the computational efficiency of the proposed method is a key aspect for autonomous vehicles. In this study, we propose GeeNet, a deep learning based method for point cloud completion and ground elevation estimation. GeeNet uses mixed 2D/3D convolutions to preserve a lightweight architecture. By using these features from the architecture, GeeNet learns the appearance, analyzes the scene, and estimates ground elevation in real time.

To address the second issue, we devise a rule-based method to acquire the ground-truth elevation map from the SemanticKITTI and SemanticPOSS datasets.

Our experiments on these solutions have shown that using the mixed 2D/3D convolutional networks to complete ground area and output ground elevation in a grid-based representation outperforms other methods. GeeNet has strong generalization ability between different datasets. GeeNet has achieved comparable performance in terms of accuracy, and new technologies have been proposed for ground point cloud completion and ground elevation estimation with a runtime of 0.88 ms.

The main contributions can be summarized as follows:

1. We propose a novel deep neural network called GeeNet for real-time point cloud completion and ground elevation estimation. GeeNet operates on 3D voxels and is trained in an end-to-end manner. It uses a mixture of 2D/3D convolutions to preserve a lightweight architecture and to regress the ground elevation information of each unit in the grid. It can predict a dense representation of the ground with clear road boundaries.

2. The process of generating ground-truth datasets for evaluating ground elevation from the SemanticKITTI and SemanticPOSS datasets is demonstrated experimentally.

3. Comprehensive experiments and comparisons are performed to reveal the robustness, efficiency, and generalization capability of our learning-based GeeNet.

2 Related works

In this section, we first introduce methods related to deep learning for point cloud semantic segmentation. Then, we discuss methods that focus on semantic scene completion. Finally, we cover methods for ground height estimation.

2.1 Point cloud semantic segmentation

Different from 2D images with regular pixels, point clouds are always sparse and disordered, and thus point cloud processing is a challenging task. Specifically, there are three main branches to tackle this task: (1) Projection-based approaches. These approaches project point clouds onto 2D pixels to apply the traditional convolutional neural network (CNN). Previous works projected all points onto 2D images by plane projection (Boulch et al., 2017; Lawin et al., 2017; Tatarchenko et al., 2018) or spherical projection (Wu BC et al., 2018, 2019). (2) Voxel-based approaches. Considering the sparse nature of point clouds and the memory consumption of operation on the point clouds, it is not effective to directly voxelize point clouds and employ 3D convolution for feature learning. Therefore, several improved methods have been proposed, including efficient spatial sparse convolution (Graham et al., 2018; Choy et al., 2019) and octree-based CNNs (Riegler et al., 2017; Wang PS et al., 2017). (3) Point-based approaches. These approaches directly process raw point clouds (Qi et al., 2017a, 2017b) using sampling strategies to select sub-points and employing local grouping with feature aggregation for local feature learning. Among these methods, graph-based learning (Landrieu and Simonovsky, 2018; Landrieu and Boussaha, 2019; Wang L et al., 2019) and convolution-like operations (Thomas et al., 2019; Wu WX et al., 2019) are widely used. However, due to the complex network architecture, previous methods often suffer from a heavy computational burden.

2.2 Semantic scene completion

Semantic scene completion tends to obtain a complete 3D voxel representation using incomplete input. Pioneered by the end-to-end model SSCNet (Song et al., 2017), single-view depth can be used as the input to output semantic labels and complete the scene simultaneously. However, SSCNet is rarely applied in large-scale LiDAR scenarios, due to severe geometric detail loss and real-time requirements. More recently, S3CNet (Cheng et al., 2021) uses a neural network based on sparse convolution to predict semantically completed scenes from a single unified LiDAR point cloud. Moreover, JS3C-Net (Yan et al., 2021) uses a single sweep LiDAR point cloud semantic segmentation framework based on contextual shape priors of the semantic scene completion network. Unlike the previous methods, local deep implicit functions (Rist et al., 2022) predict a continuous scene representation that is not based on voxelization; instead, a network is devised to complete the navigable areas with low computational complexity and potential of use in real-time applications.

2.3 Ground elevation estimation

The early method for ground elevation estimation projects 3D points as a 2.5D grid and then applies min–max elevation maps used in the Defense Advanced Research Projects Agency (DARPA) Urban Challenge (Thrun et al., 2006). However, this method struggles with significant errors in the corner case of bridges and treetops. Another group of approaches are based on a 2D-line-extraction-based fast algorithm (Himmelsbach et al., 2010), but they face issues with scalability to a large set of cases. Other methods leverage the gradient information of the terrain to model the ground plane with the help of a Markov random field (MRF) or a conditional random field (CRF) (Byun et al., 2015). More recently, a two-stage work was proposed to estimate the ground points and then fit a plane by RANSAC (Narksri et al., 2018). Also, Gaussian process regression (GPR) and robust locally weighted regression (RLWR) have been integrated to build a hybrid regression model for the ground plane, but this is not a real-time approach because of the high computational complexity of the Gaussian processes (Liu KQ et al., 2019). Most of the methods discussed take

advantage of hand-crafted features that are often complex to implement, computationally expensive, and not scalable. The sparsity, occlusions, and roughness of the ground are not taken into consideration, which leads to unsatisfactory results.

3 Methods

We regard the ground elevation estimation problem as a point cloud completion issue, where the height information can be obtained from the complete output of the network by taking a single sparse LiDAR sweep as the input. Then, we cope with the issue of dense 3D semantic completion by assigning a semantic label to each individual voxel. When given a sparse 3D grid from incomplete input, the target is to predict the 3D semantic scene representation, in which each cell is assigned a semantic label $C = [c_0, c_1]$, where c_0 represents free voxels and c_1 represents ground voxels.

The architecture of GeeNet (Fig. 2) leverages a lightweight U-Net-like architecture to infer 3D semantic completion, which is favorable to autonomous vehicle applications. Apart from 3D convolutions with high computational complexity, the 2D convolutions are used mostly along the height axis close to a bird's eye view (BEV). In the following subsections, we will systematically introduce the customized lightweight architecture, reconstruction, post-processing, and training details.

3.1 Lightweight architecture

Aiming to predict a dense output from the sparse input, an encoder–decoder U-Net-like backbone is leveraged to learn features at decreasing resolutions. A set of convolutions are employed, followed by a pooling that downscales the resolution size by 2. Reducing spatial dimensions in U-Nets is beneficial for semantic tasks because it increases the kernels' field of view at no cost. The dense convolutions in the encoder mean dilation of the input manifold, which is helpful in 3D semantic completion with the sparse to dense nature.

3.1.1 Two-dimensional pipeline

To preserve the lightweight architecture, 2D convolutions are used along the X and Y dimensions, with the height dimension (Z) as a feature dimension. Note that the 3D data are processed directly, different from other 2D/3D methods that rely on 2.5D data, such as depth and BEV. Although 2D convolutions lead to losing 3D spatial connectedness, they also realize remarkably lighter operations. Further, a minimum number of features are maintained in each convolution layer to reduce the memory requirements. Moreover, the standard skip connections are employed to enhance information flow by concatenating the output of each level to all lower levels in the decoder. Specifically, coarse feature maps are upsampled by learning ad-hoc deconvolution before concatenation to lower levels, shown as blue deconvolution modules and skip-connection in Fig. 2. Therefore, this operation encourages GeeNet to use high-level information from coarser resolutions, which boosts spatial contextual information.

3.1.2 Three-dimensional segmentation head

Different from previous methods dealing with point clouds from a BEV, for 3D ground completion we need to regain the third dimension with 2D convolutions. Specifically, the decoder must output a 4D tensor while 2D CNNs output 3D feature maps, where the last dimension is the semantic class-wise probability distribution. To tackle this, as illustrated in the gray module in Fig. 2, 3D segmentation heads are introduced which leverage a series of dense and dilated convolutions. In detail, the dilated convolutions, which can be regarded as atrous spatial pyramid pooling (ASPP) (Chen LC et al., 2017), are beneficial for combining information from different receptive fields based on the merits of the convolutions with increasing dilation rates. Note that although dilated convolution is light and powerful, it is not suitable for sparse inputs and therefore cannot be applied on encoders. In the segmentation head, the advantages of preceding ASPP with dense 3D convolutions are twofold: It can not only improve dense feature mapping but also separate the segmentation features and backbone features.

3.1.3 Ground completion

In this module, the goal of the network is to perform ground completion more quickly to help with the ground elevation estimation for mobile robotics applications and autonomous vehicles. We attach a 3D segmentation head after the 2D U-Net backbone. The sample output is shown in Fig. 2. In

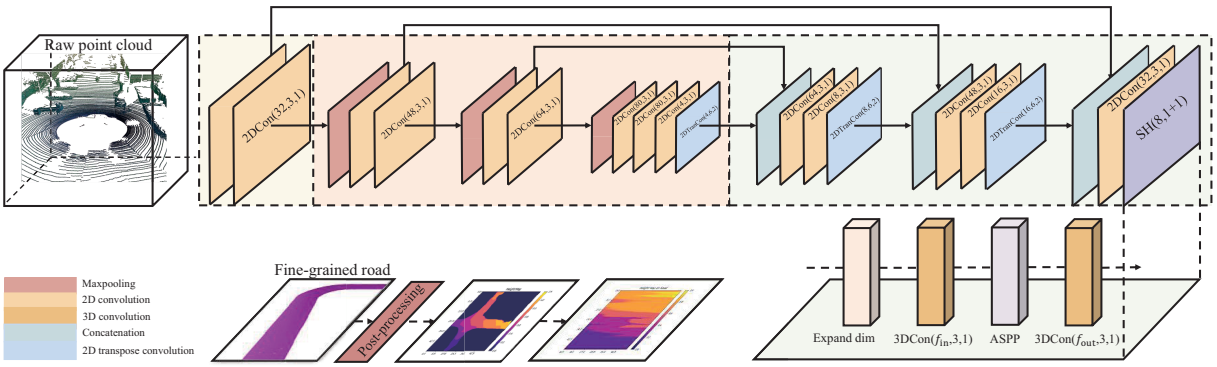

Fig. 2 GeeNet architecture. The pipeline of GeeNet leverages a U-Net along with 2D backbone convolutions (in yellow) and 3D segmentation heads (in purple) to fulfill 3D semantic segmentation and completion with low complexity. Note that convolution parameters are listed as the number of filters, kernel size, and stride. Notice that the dimension of the 2D features is intentionally reduced, and atrous 3D convolutions are integrated to maintain low inference complexity. References to color refer to the online version of this figure

experiments, we notice that the need to ward off features from the segmentation heads caused by the main features of the 2D backbone justifies the additional 3D convolutions in the segmentation head. The key interest of the architecture is that it can infer ground completion while reducing the computation and memory requirements.

3.1.4 Post-processing

As the network outputs are multi-layer and contain a small amount of noise, post-processing is performed to obtain fine-grained results. Inspired by Narksri et al. (2018), RANSAC is leveraged to denoise the noise grids, which are used for segments of a plane in the point cloud. The distance threshold is typically set to 0.3 m; i.e., if the distance from a point to the targeted plane is no larger than 0.3 m, the point is considered an inlier. The number of initial points is set to 2000, which represent inliers in each iteration; the number of iterations is 1000. After RANSAC, noise is removed from the ground grids. Furthermore, the top layer of the grids will be maintained to obtain the final ground areas.

3.2 Training

GeeNet is trained from scratch in an end-to-end manner, with pairs of sparse input voxels (x) and semi-dense semantically labeled voxel grid ($\hat{y}$). Note that densely labeled ground truth is impractical for scene completion in the real setup because of the occlusions and sensor's field-of-view limitations. Therefore, the ground truth $\hat{y}$ is encoded with two classes (one road class and one free class). Inspired by Song et al. (2017), Liu S et al. (2018), and Garbade et al. (2019), a sparse loss strategy is used to back-propagate the gradient only when the ground truth is known. The cross-entropy loss employed to train the network is defined as

$$\mathcal{L} = -\sum_{c=0}^{N} w_c \hat{y}_{i,c} \log\left(\frac{\mathrm{e}^{y_{i,c}}}{\sum_{c=0}^{N} \mathrm{e}^{y_{i,c}}}\right), \qquad (1)$$

where y denotes the network output, i represents the index of voxels, and $\hat{y}_{i,c}$ is a one-hot vector ($\hat{y}_{i,c}=1$ if voxel i is labeled as class c; otherwise, $\hat{y}_{i,c}=0$). $w_c = \frac{1}{\log(f_c+\epsilon)}$ (with $\epsilon \ll 1$) is leveraged by weighting class loss based on the inverse of class frequency f_c for the imbalance between the semantic road class and free class. Note that some of our choices aim at faster training or higher inference speed.

4 Experiments

4.1 Datasets

GeeNet is trained and evaluated on the widely used semantic scene completion benchmarks SemanticKITTI (Behley et al., 2019) and SemanticPOSS (Pan et al., 2020), which provide 3D voxel grids from semantically labeled scans of HDL-64E rotating LiDAR in outdoor urban scenes. SemanticKITTI is the largest LiDAR sequence dataset with point-level annotations, including 43 552 densely annotated LiDAR scans in 21 sequences. A total of 19 valid categories are annotated

in these scans, and each scan spans up to 160 m × 160 m × 20 m, with more than 10^5 points. SemanticPOSS is a newly proposed dataset with 11 annotated categories similar to SemanticKITTI, but even more challenging is that each scene contains more than 20 times as many sparse small objects (i.e., people and bicycles), while the total number of frames is only 1/20 of SemanticKITTI's.

4.2 Preprocessing

Following Behley et al. (2019), network inputs are voxelized single scans, while the ground truths are from the voxelized aggregation of successive scans. We require the voxel grids that belong only to the road in our method. Therefore, the inputs are processed to maintain the road class, and other classes are set as free classes. There are $256\times256\times32$ grids with 0.2 m voxel size, and the input and ground truth are sparse with an average density of 6.7% and 65.8%, respectively. Note that ground completion is essential for height information extraction in autonomous vehicles. Because there are many grid cells without any ground points, we explore rule-based methods to tackle this issue.

4.3 Rule-based methods

Many cells belonging to the ground do not have any point inside, leading to the elevation information of the cell to be zero. Therefore, we concatenate the continuous 50 frames, average the z values of all the ground points in the corresponding cells, and create an elevation map (Fig. 3). Our goal is to achieve a smooth and uniform ground plane elevation, even in sparse points and occluded areas. We use an image inpainting technique to fill in the holes in the elevation map.

4.4 Baselines

In this subsection, we introduce the differences among five baselines from several aspects: method, improvement, dataset, and architecture. Table 1 shows the differences among baselines on the dataset and architecture.

SSCNet (Song et al., 2017) aims to establish a model for scene completion and scene labeling from a single depth image of a 3D scene. The architecture of SSCNet is based on an end-to-end 3D convolutional network to predict the final voxel labels. SSCNet constructs an end-to-end 3D convolutional network using a single depth image as the input and adopts an extended 3D context module to achieve 3D context learning. Its output is four times downsampled.

SSCNet-full (Song et al., 2017) is the SSCNet using deconvolution to achieve full input resolution.

JS3C-Net (Yan et al., 2021) is focused on solving the problems of shape loss and uneven point sampling caused by its sparsity in semantic scene segmentation. Its core architecture is based on a novel sparse semantic segmentation framework. The idea is to construct a semantic segmentation network based on point clouds, which is assisted by contextual shape priors. In addition, a disposable point-voxel interaction module is proposed for knowledge fusion. To improve the semantic completion performance, dense upsampling is used to replace SSCNet's upsampling based on dilation convolution, thereby avoiding inaccurate interpolation. In addition, research in large-scale LiDAR scenarios is conducted to address the missing geometric details and real-time requirements of SSCNet.

LMSCNet (Roldão et al., 2020) aims to solve the extensive computation problem caused by predicted resolution and network depth limitations in semantic scene completion. The architecture of LMSCNet is based on a lightweight U-Net architecture with 2D backbone convolution and 3D segmentation heads to predict multi-scale 3D semantic completion. LMSCNet uses a single sparse LiDAR scan as the input to construct a lightweight multi-scale semantic completion network for fast inference. To improve the semantic completion performance, LMSCNet adopts a sparse loss strategy like SSCNet, but it avoids the use of truncated signed distance function (TSDF) preprocessing, thereby reducing the inference time.

4.5 Implementation details

We follow the training split protocol with 3834/815 grids of Roldão et al. (2020), using $(x-y)$ flipping augmentation for generalization. We use the Adam optimizer ($\beta_1=0.9, \beta_2=0.999$) with a learning rate of 0.001 scaled by $0.98^{n_{\text{epoch}}}$, where n_{epoch} represents the number of times the neural network is trained. GeeNet is trained on a single 32 GB GPU with a batch size of 8 for 80 epochs.

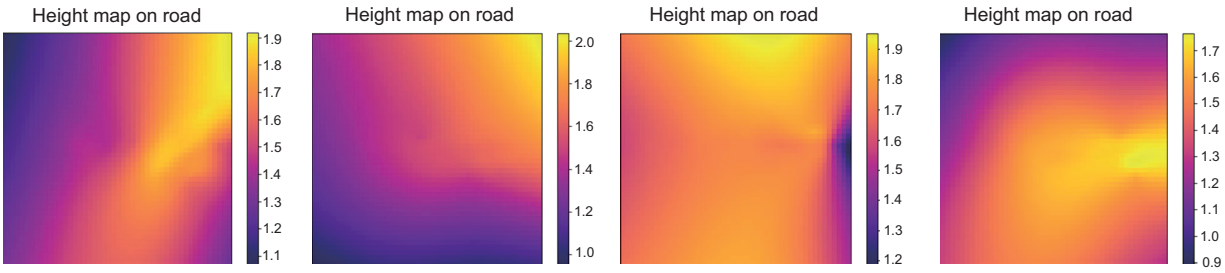

Fig. 3 Qualitative results of the ground-truth elevation map generated using the rule-based method (in meters)

Table 1 Comparison of differences among baselines on the dataset and architecture

Method	Dataset	Architecture
SSCNet	NYU SUNCG	3D convolutional network
JS3C-Net	SemanticPOSS SemanticKITTI	Sparse convolution U-Net Dense convolution neural network Point-voxel interaction
LMSCNet	SemanticKITTI	2D backbone convolutional network 3D segmentation head
Rule-based		Concatenating the continuous frames and averaging the z values of all the ground points

4.6 Metrics

We use the Chamfer distance and F-score as our evaluation metrics for point cloud completion and ground elevation estimation. The Chamfer distance can measure the distance between the set of predicted ground grid centers P and the ground-truth ground grid centers G, computed as follows:

$$\begin{aligned}\mathrm{CD}(P,G)=&\frac{1}{\mid P\mid}\sum_{p\in P}\min_{g\in G}\|p-g\|\\&+\frac{1}{\mid G\mid}\sum_{g\in G}\min_{p\in P}\|g-p\|.\end{aligned}\tag{2}$$

CD-ℓ_1 with ℓ_1-normalization and CD-ℓ_2 with ℓ_2-normalization are leveraged as the evaluation metrics to calculate the distance between two point clouds. Moreover, the mean intersection-over-union (IoU) over two classes is used on the ground and as a metric for point cloud completion. The precision and recall values are also provided. GeeNet shows results superior to those of other methods.

$$\mathrm{IoU}=\frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FP}_c+\mathrm{FN}_c},\tag{3}$$

where TP_c, FP_c, and FN_c correspond to the numbers of true positive, false positive, and false negative predictions for road class c, respectively.

5 Experimental results

In this section, the GeeNet performance is compared with those of three state-of-the-art methods: JS3C-Net (Yan et al., 2021), LMSCNet (Roldão et al., 2020), and SSCNet (Song et al., 2017). Due to the output of SSCNet being four times downsampled, the method of using deconvolution to obtain full input resolution is also evaluated, denoted as SSCNet-full (Song et al., 2017). The semantic road completion and ground height are detailed. Moreover, the speed and lightness of GeeNet are demonstrated.

5.1 Performance on road completion

To evaluate GeeNet, the experiments on point cloud completion for road areas were systematically carried out. The results of road completion have significant influence on ground elevation estimation. The quantitative evaluations on the SemanticKITTI dataset are shown in Table 2, from which we can see that GeeNet outperformed JS3C-Net and LMSCNet by 32.6% and 29.8% in terms of CD-ℓ_2, respectively,

demonstrating that the set of grid centers by GeeNet was much closer to the ground truth. Moreover, GeeNet was superior to the other methods in terms of IoU, precision, recall, and F-score@1%. Further, the quality comparison of ground completion is shown in Fig. 4. Note that GeeNet and LMSCNet were apparently superior to others. However, the performance of LMSCNet was not satisfactory, because the details were not reasonable enough, which can be seen from the ellipses in Fig. 4. On the other hand, GeeNet completed more detailed ground areas. Furthermore, the ground completion experiments were conducted on the SemanticPOSS dataset. As shown in Table 3, GeeNet achieved the best CD-ℓ_1, CD-ℓ_2,

Table 2 Quantitative evaluation on point cloud completion of the road on the SemanticKITTI dataset

Method	CD-ℓ_2 ($\times 10^3$)	CD-ℓ_1 ($\times 10^2$)	IoU	Precision	Recall	F-score@1%
JS3C-Net	1.29	0.69	60.75	66.59	77.39	71.36
LMSCNet	1.24	0.62	70.04	83.65	76.95	82.38
SSCNet-full	3.02	3.50	25.45	25.81	64.83	40.57
SSCNet	2.40	1.79	21.62	21.84	65.66	35.56
GeeNet	**0.87**	**0.43**	**71.07**	**85.45**	**80.85**	**83.09**

The bold font denotes the best performance

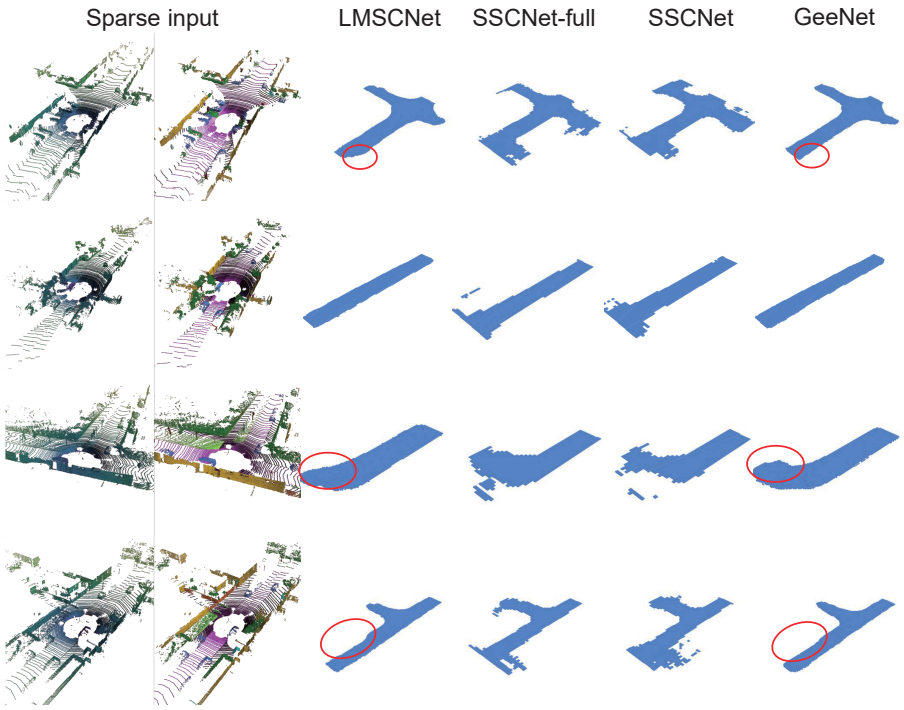

Fig. 4 Comparison of point cloud completion of the road on the SemanticKITTI dataset. The ellipses indicated that GeeNet performed better than LMSCNet, SSCNet-full, and SSCNet

Table 3 Quantitative evaluation on point cloud completion of the road on the SemanticPOSS dataset

Method	CD-ℓ_2 ($\times 10^3$)	CD-ℓ_1 ($\times 10^2$)	IoU	Precision	Recall	F-score@1%
JS3C-Net	1.65	1.09	55.90	60.11	70.25	67.30
LMSCNet	1.87	1.28	61.34	78.50	72.39	76.21
SSCNet-full	3.89	4.02	22.22	23.19	58.32	36.56
SSCNet	3.35	2.44	20.23	20.39	60.66	33.28
GeeNet	**1.33**	**0.88**	**62.21**	**80.33**	**78.59**	**79.91**

The bold font denotes the best performance

and IoU, showing that GeeNet performed the best across various datasets.

5.2 Performance on ground elevation

The performances of GeeNet and other methods on ground elevation were also evaluated. The ground truth came from our proposed rule-based method, which concatenated 50 consecutive frames, filtered the points outside the road, and then interpolated them to obtain height maps of the ground. First, we obtained the height maps from the output of point cloud completion, as shown in Fig. 5. It can be seen that a complete ground with precise road bounds was produced, without any holes or noises on the road. Then, we removed the areas that did not belong to the road and interpolated them to obtain the height maps. These height maps will be evaluated with the ground truth from the rule-based method. The results on the SemanticKITTI and SemanticPOSS datasets are compared in Tables 4 and 5, respectively, where GeeNet had the lowest CD-ℓ_1 and CD-ℓ_2, exhibiting the best height elevation performance.

Table 4 Quantitative evaluation of ground elevation estimation on the SemanticKITTI dataset

Method	CD-ℓ_2 ($\times 10^2$)	CD-ℓ_1
JS3C-Net	2.14	0.44
LMSCNet	1.99	0.47
SSCNet-full	2.63	0.54
SSCNet	2.32	0.49
GeeNet	**1.79**	**0.38**

The bold font denotes the best performance

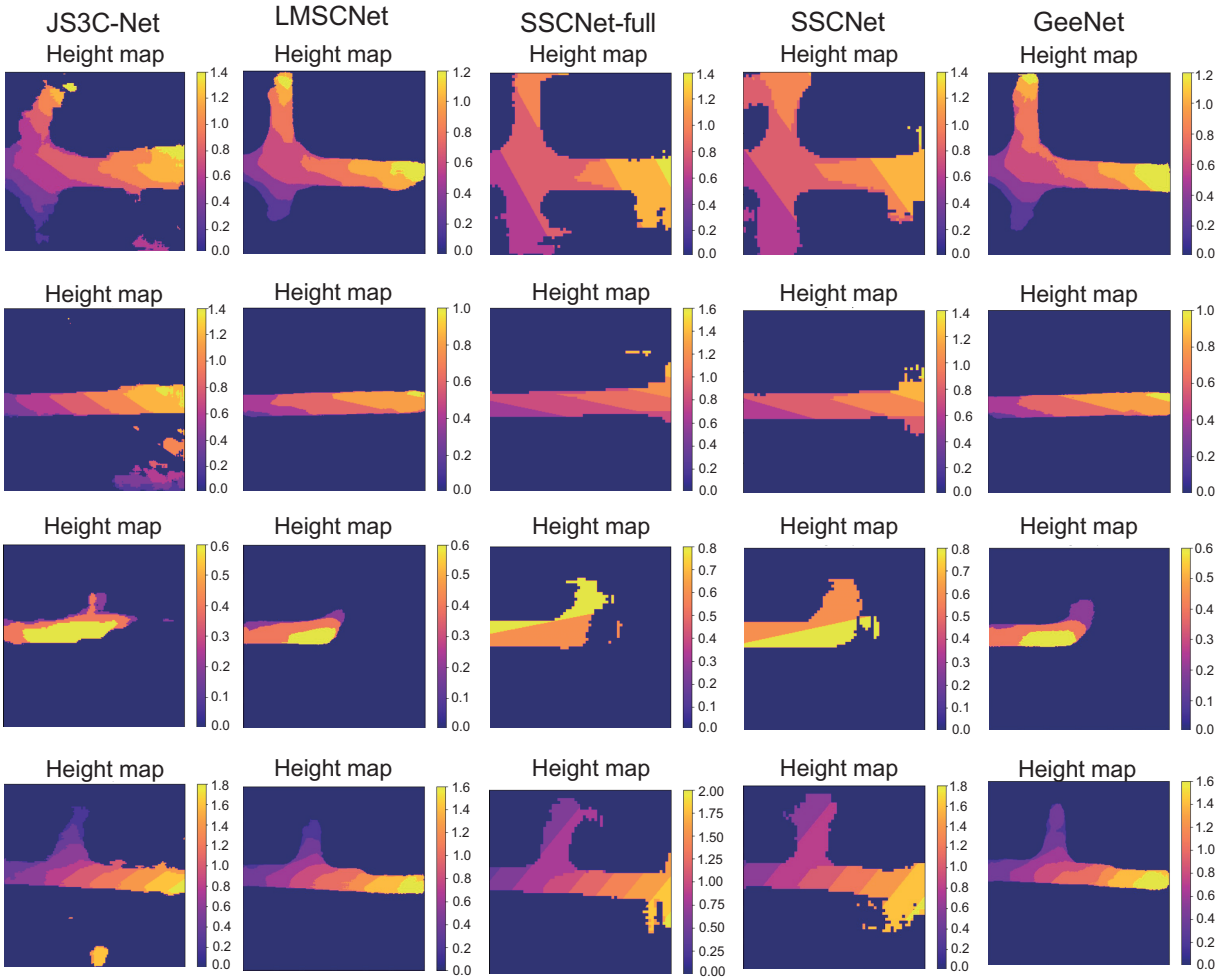

Fig. 5 Comparison of ground elevation estimation on the SemanticKITTI dataset (in meters). GeeNet fulfilled the best ground elevation estimation from the bird's eye view

Table 5 Quantitative evaluation of ground elevation estimation on the SemanticPOSS dataset

Method	CD-ℓ_2 ($\times 10^2$)	CD-ℓ_1
JS3C-Net	3.45	1.33
LMSCNet	3.02	1.08
SSCNet-full	3.88	1.82
SSCNet	3.75	1.58
GeeNet	**2.66**	**0.89**

The bold font denotes the best performance

5.3 Generalization experiments

To evaluate whether scene diversity has impact on the performance of deep learning models, we conducted generalization experiments using GeeNet. The results are shown in Table 6. For simplicity, we denote the training of GeeNet on SemanticKITTI as GeeNet-KITTI and the training of GeeNet on SemanticPOSS as GeeNet-POSS. From the experimental results, we found that the IoU of GeeNet-KITTI for SemanticPOSS was dropped by 12.28, whereas the IoU of GeeNet-POSS for SemanticKITTI was dropped by 7.33. Note that the IoU of GeeNet-KITTI on SemanticPOSS was only 3.42 lower than that of GeeNet-POSS, showing the good generalization performance of GeeNet.

Table 6 Cross-dataset generalization experiment between SemanticPOSS and SemanticKITTI by GeeNet

Method	IoU	
	SemanticKITTI	SemanticPOSS
GeeNet-KITTI	71.07	58.79
GeeNet-POSS	54.88	62.21

GeeNet trained on SemanticKITTI is denoted as GeeNet-KITTI, while GeeNet trained on SemanticPOSS is denoted as GeeNet-POSS

5.4 Analysis of runtime

To gain insight into the potential use of GeeNet in a real-time setting, further analysis of the inference time was carried out, as shown in Table 7. GeeNet is a lightweight network with a runtime of 0.88 ms compared to 63.88 s of the rule-based method in predicting the ground height in the $256 \times 256 \times 32$ grid. Note that GeeNet outperformed JS3C-Net, LMSCNet, SSCNet, and SSCNet-full by a great margin, proving the strong inference ability on the ground elevation estimation of GeeNet and the possible loading on autonomous vehicles.

Table 7 Computational time required by GeeNet and other methods

Method	Time	Device
Rule-based	63.88 s	CPU
JS3C-Net	0.91 ms	GPU*
LMSCNet	0.99 ms	GPU*
SSCNet-full	1.04 ms	GPU*
SSCNet	1.00 ms	GPU*
GeeNet	0.88 ms	GPU*

* RTX 3090

5.5 Analysis of architectures

To gain insight into the superiority of GeeNet, Table 8 compares the network statistics of GeeNet and baseline methods. It can be seen that GeeNet required the fewest parameters (0.33×10^6) at the lowest computational cost for inference (27.3 Gflops), superior to JS3C-Net, LMSCNet, SSCNet-full, and SSCNet. Moreover, GeeNet achieved 89.30 frames per second (FPS) during inference, much faster than other methods, showing its potential of use in real-time applications.

Table 8 Comparison of network statistics among GeeNet and other methods

Method	Number of parameters ($\times 10^6$)	Gflops	FPS
JS3C-Net	2.70	189.8	60.33
LMSCNet	0.35	72.6	21.28
SSCNet-full	1.09	769.6	45.94
SSCNet	0.93	82.5	56.90
GeeNet	**0.33**	**27.3**	**89.30**

The bold font denotes the best performance

5.6 Ablation studies

To unveil the effectiveness of each module in GeeNet, comprehensive ablation studies were carried out on the SemanticKITTI validation set. Three models were devised for comparison: model A replaces the upsampling layers with deconvolution layers, model B removes the ASPP module, and model C takes vanilla U-Net instead of our hierarchical U-Net.

5.6.1 Effect of dilated convolutions

In GeeNet, the parameter-greedy deconvolution layers were replaced with upsampling layers to preserve the lightweight architecture. As shown in Table 9, model A with deconvolution layers achieved

6.17 drops on the IoU of the road, proving the effectiveness of upsampling layers.

5.6.2 Effect of upsampling

As we can see from Table 9, model B without ASPP resulted in a 3.52 decrement in IoU, indicating that the increasing receptive fields of the inner dilated convolutions from ASPP can bring richer features.

5.6.3 Hierarchical U-Net decoder

The ablation studies on the hierarchical U-Net decoder were carried out, where the hierarchical U-Net decoder was replaced with vanilla U-Net in model C. As seen in Table 9, GeeNet improved the performance of model C from 68.42 to 71.07. Benefiting from the hierarchical U-Net, GeeNet aggregates hierarchical features and considers coarser semantic features for fine resolutions (Fig. 6).

Table 9 Module-level ablation studies on the SemanticKITTI dataset

Model	Upsampling	ASPP	Hierarchical U-Net	IoU
A		√	√	64.90
B	√		√	67.55
C	√	√		68.42
GeeNet	√	√	√	71.07

ASPP: atrous spatial pyramid pooling

6 Conclusions

This paper tackled the challenging issues of point cloud completion and ground elevation estimation for autonomous vehicles. We proposed a robust and fast deep learning based approach called GeeNet, using mixed 2D/3D convolution networks to complete the ground area and output ground elevation in a grid-based representation. The qualitative and quantitative evaluations of GeeNet on the SemanticKITTI and SemanticPOSS datasets demonstrated the superiority of GeeNet to other methods. Further, GeeNet had great generalization ability across different datasets. GeeNet achieved comparable performance in terms of point cloud completion and ground elevation estimation, with a runtime of 0.88 ms.

Contributors

Ben FEI designed the research. Liwen LIU processed the data and drafted the paper. Weidong YANG and Ben FEI revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Behley J, Garbade M, Milioto A, et al., 2019. SemanticKITTI: a dataset for semantic scene understanding of LiDAR sequences. Proc IEEE/CVF Int Conf on Computer Vision, p.9296-9306. https://doi.org/10.1109/ICCV.2019.00939

Boulch A, le Saux B, Audebert N, 2017. Unstructured point cloud semantic labeling using deep segmentation

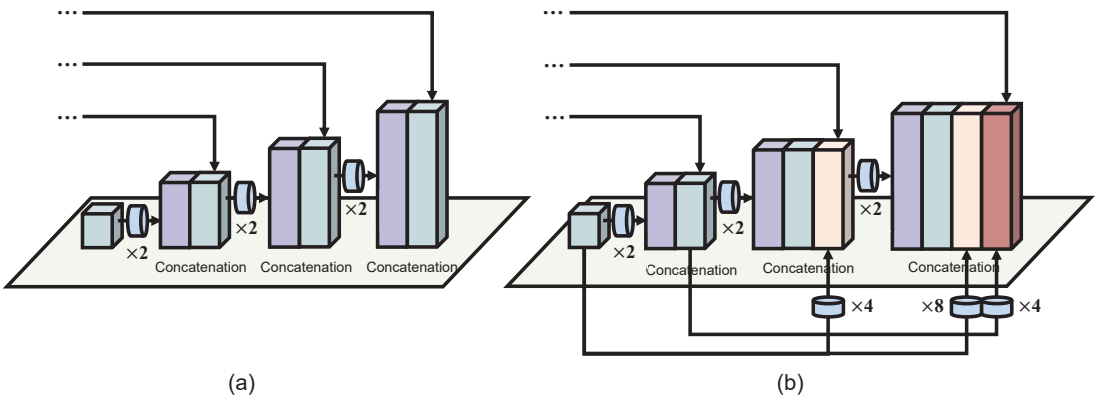

Fig. 6 Architectures of the vanilla U-Net decoder (a) and hierarchical decoder (b)

networks. Proc Workshop on 3D Object Retrieval, p.17-24. https://doi.org/10.2312/3dor.20171047

Byun J, Na KI, Seo BS, et al., 2015. Drivable road detection with 3D point clouds based on the MRF for intelligent vehicle. In: Mejias L, Corke P, Roberts J (Eds.), Field and Service Robotics. Springer, Cham, Switzerland, p.49-60. https://doi.org/10.1007/978-3-319-07488-7_4

Chen LC, Papandreou G, Kokkinos I, et al., 2017. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans Patt Anal Mach Intell*, 40(4):834-848. https://doi.org/10.1109/TPAMI.2017.2699184

Chen YD, Hao CY, Wu W, et al., 2016. Robust dense reconstruction by range merging based on confidence estimation. *Sci China Inform Sci*, 59(9):092103. https://doi.org/10.1007/s11432-015-0957-4

Cheng R, Agia C, Ren Y, et al., 2021. S3CNet: a sparse semantic scene completion network for LiDAR point clouds. Proc Conf on Robot Learning, p.2148-2161.

Choy C, Gwak J, Savarese S, 2019. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3070-3079. https://doi.org/10.1109/CVPR.2019.00319

Garbade M, Chen YT, Sawatzky J, et al., 2019. Two stream 3D semantic scene completion. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition Workshops, p.416-425. https://doi.org/10.1109/CVPRW.2019.00055

Graham B, Engelcke M, van der Maaten L, 2018. 3D semantic segmentation with submanifold sparse convolutional networks. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9224-9232. https://doi.org/10.1109/CVPR.2018.00961

Himmelsbach M, Hundelshausen FV, Wuensche HJ, 2010. Fast segmentation of 3D point clouds for ground vehicles. IEEE Intelligent Vehicles Symp, p.560-565. https://doi.org/10.1109/IVS.2010.5548059

Landrieu L, Boussaha M, 2019. Point cloud oversegmentation with graph-structured deep metric learning. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.7432-7491. https://doi.org/10.1109/CVPR.2019.00762

Landrieu L, Simonovsky M, 2018. Large-scale point cloud semantic segmentation with superpoint graphs. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4558-4567. https://doi.org/10.1109/CVPR.2018.00479

Lang AH, Vora S, Caesar H, et al., 2019. PointPillars: fast encoders for object detection from point clouds. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.12689-12697. https://doi.org/10.1109/CVPR.2019.01298

Lawin FJ, Danelljan M, Tosteberg P, et al., 2017. Deep projective 3D semantic segmentation. Proc 17th Int Conf on Computer Analysis of Images and Patterns, p.95-107. https://doi.org/10.1007/978-3-319-64689-3_8

Leonard J, How J, Teller S, et al., 2008. A perception-driven autonomous urban vehicle. *J Field Robot*, 25(10):727-774. https://doi.org/10.1002/rob.20262

Liu BS, Chen XM, Han YH, et al., 2019. Accelerating DNN-based 3D point cloud processing for mobile computing. *Sci China Inform Sci*, 62(11):212206. https://doi.org/10.1007/s11432-019-9932-3

Liu KQ, Wang WG, Tharmarasa R, et al., 2019. Ground surface filtering of 3D point clouds based on hybrid regression technique. *IEEE Access*, 7:23270-23284. https://doi.org/10.1109/ACCESS.2019.2899674

Liu S, Hu Y, Zeng YM, et al., 2018. See and think: disentangling semantic scene completion. Proc 32nd Int Conf on Neural Information Processing Systems, p.261-272.

Liu Y, 2016. Robust segmentation of raw point clouds into consistent surfaces. *Sci China Technol Sci*, 59(8):1156-1166. https://doi.org/10.1007/s11431-016-6072-8

Narksri P, Takeuchi E, Ninomiya Y, et al., 2018. A slope-robust cascaded ground segmentation in 3D point cloud for autonomous vehicles. Proc 21st Int Conf on Intelligent Transportation Systems, p.497-504. https://doi.org/10.1109/ITSC.2018.8569534

Paigwar A, Erkent Ö, Sierra-Gonzalez D, et al., 2020. GndNet: fast ground plane estimation and point cloud segmentation for autonomous vehicles. IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.2150-2156. https://doi.org/10.1109/IROS45743.2020.9340979

Pan YC, Gao B, Mei JL, et al., 2020. SemanticPOSS: a point cloud dataset with large quantity of dynamic instances. IEEE Intelligent Vehicles Symp, p.687-693. https://doi.org/10.1109/IV47402.2020.9304596

Qi CR, Su H, Mo KC, et al., 2017a. PointNet: deep learning on point sets for 3D classification and segmentation. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.77-85. https://doi.org/10.1109/CVPR.2017.16

Qi CR, Yi L, Su H, et al., 2017b. PointNet++: deep hierarchical feature learning on point sets in a metric space. Proc 31st Int Conf on Neural Information Processing Systems, p.5105-5114.

Ren DY, Wu ZY, Li JW, et al., 2022. Point attention network for point cloud semantic segmentation. *Sci China Inform Sci*, 65(9):192104. https://doi.org/10.1007/s11432-021-3387-7

Riegler G, Ulusoy AO, Geiger A, 2017. OctNet: learning deep 3D representations at high resolutions. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.6620-6629. https://doi.org/10.1109/CVPR.2017.701

Rist CB, Emmerichs D, Enzweiler M, et al., 2022. Semantic scene completion using local deep implicit functions on LiDAR data. *IEEE Trans Patt Anal Mach Intell*, 44(10):7205-7218. https://doi.org/10.1109/TPAMI.2021.3095302

Roldão L, de Charette R, Verroust-Blondet A, 2020. LMSCNet: lightweight multiscale 3D semantic completion. Int Conf on 3D Vision, p.111-119. https://doi.org/10.1109/3DV50981.2020.00021

Rummelhard L, Nègre A, Laugier C, 2015. Conditional Monte Carlo dense occupancy tracker. IEEE 18th Int Conf on Intelligent Transportation Systems, p.2485-2490. https://doi.org/10.1109/ITSC.2015.400

Rummelhard L, Paigwar A, Nègre A, et al., 2017. Ground estimation and point cloud segmentation using spatiotemporal conditional random field. IEEE Intelligent Vehicles Symp, p.1105-1110. https://doi.org/10.1109/IVS.2017.7995861

Song SR, Yu F, Zeng A, et al., 2017. Semantic scene completion from a single depth image. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.190-198. https://doi.org/10.1109/CVPR.2017.28

Tatarchenko M, Park J, Koltun V, et al., 2018. Tangent convolutions for dense prediction in 3D. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3887-3896. https://doi.org/10.1109/CVPR.2018.00409

Thomas H, Qi CR, Deschaud JE, et al., 2019. KPConv: flexible and deformable convolution for point clouds. Proc IEEE/CVF Int Conf on Computer Vision, p.6410-6419. https://doi.org/10.1109/ICCV.2019.00651

Thrun S, Montemerlo M, Dahlkamp H, et al., 2006. Stanley: the robot that won the DARPA grand challenge. *J Field Robot*, 23(9):661-692. https://doi.org/10.1002/rob.20147

Wang L, Huang YC, Hou YL, et al., 2019. Graph attention convolution for point cloud semantic segmentation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.10288-10297. https://doi.org/10.1109/CVPR.2019.01054

Wang PS, Liu Y, Guo YX, et al., 2017. O-CNN: octree-based convolutional neural networks for 3D shape analysis. *ACM Trans Graph*, 36(4):72. https://doi.org/10.1145/3072959.3073608

Wu BC, Wan A, Yue XY, et al., 2018. SqueezeSeg: convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. IEEE Int Conf on Robotics and Automation, p.1887-1893. https://doi.org/10.1109/ICRA.2018.8462926

Wu BC, Zhou XY, Zhao SC, et al., 2019. SqueezeSegV2: improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud. Int Conf on Robotics and Automation, p.4376-4382. https://doi.org/10.1109/ICRA.2019.8793495

Wu WX, Qi ZA, Li FX, 2019. PointConv: deep convolutional networks on 3D point clouds. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9613-9622. https://doi.org/10.1109/CVPR.2019.00985

Yan X, Gao JT, Li J, et al., 2021. Sparse single sweep LiDAR point cloud segmentation via learning contextual shape priors from scene completion. Proc AAAI Conf on Artificial Intelligence, p.3101-3109. https://doi.org/10.48550/arXiv.2012.03762

Zhou Y, Tuzel O, 2018. VoxelNet: end-to-end learning for point cloud based 3D object detection. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4490-4499. https://doi.org/10.1109/CVPR.2018.00472