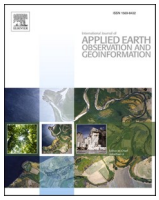
International Journal of Applied Earth Observations and Geoinformation 112 (2022) 102904

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag



IDA-Net: Intensity-distribution aware networks for semantic segmentation of 3D MLS point clouds in indoor corridor environments

Zhipeng Luo [a], Pengxin Chen [a], Wenzhong Shi [a,*], Jonathan Li [b]

[a] Department of Land Surveying and Geo-Informatics, and Otto Poon Charitable Foundation Smart Cities Research Institute, The Hong Kong Polytechnic University, 999077, Hong Kong

[b] Department of Geography and Environmental Management and the Department of Systems Design Engineering, University of Waterloo, Waterloo, Canada

ARTICLE INFO

Keywords:
3D MLS point cloud
Semantic segmentation
Intensity-distribution aware
Two-stage embedding network

ABSTRACT

Semantic segmentation of 3D mobile laser scanning point clouds is the foundational task for scene understanding in several fields. Most existing segmentation methods tend to simply stack the common point attributes, such as the coordinates and intensity, but ignore their heterogeneous. This paper presents IDA-Net, an intensity-distribution aware network that mines the uniqueness and discrepancy of these two modalities in a separate way for point cloud segmentation under indoor corridor environments. Specifically, IDA-Net consists of two key components. Firstly, an intensity-distribution aware (IDA) descriptor is proposed to mine the intensity distribution pattern. It outputs a multi-channel mask for each point to represent the intensity distribution information. Secondly, a two-stage embedding network is designed to fuse the coordinates and intensity information efficiently. It includes a guiding operation in training stage and a refining operation in testing stage. IDA-Net was evaluated on two indoor corridor areas. Experimental results show that the proposed method significantly improves the performance of segmentation. Specifically, with backbone of KPConv, IDA-Net achieves high mIoU of 90.58% and 88.94% on the above two testing areas respectively, which demonstrates the superiority of the designed IDA descriptor and two-stage embedding network.

1. Introduction

As one of the foundational tasks in scene perception, scene semantic segmentation of 3D point clouds aims to label each point and understand the scene. The point-level semantic information can also be provided for downstream tasks, such as place recognition (Uy and Lee, 2018), instance segmentation (Wang et al., 2018), and scene reconstruction (Wu et al., 2020). Therefore, it plays the key role in several fields, including the automatic driving (ADs), intelligent transportation (IT), SLAM (Broggi et al., 2013; Schreiber et al., 2013; Seo et al., 2015), and so on. In this paper, we focus on scene semantic segmentation using 3D mobile laser scanning (MLS) point clouds in indoor environments.

Several methods have been undertaken for 3D point clouds semantic segmentation. Traditionally, handcraft-designed based descriptors are first used for each point to extract features, which are inputted into classical classifiers, such as the SVMs and Random forest (Guo et al., 2014). However, since handcraft-designed descriptors can just mine the shallow features, the performance of traditional methods is usually far from satisfactory (Guo et al., 2021). Recently, as one kind of technologies that has been proved powerful in mining deep features in data-driven way, deep learning (DL) has been utilized in 3D semantic segmentation. They can be broadly classified into four categories: projection-, discretization-, point- and graph-based methods (Guo et al., 2021). Projection-based and discretization-based methods firstly convert unstructured 3D point clouds into 2D image or 3D grid volumetric data, then standard 2D or 3D convolutional neural networks (CNNs) are applied to process these regular representations. Representative methods include the view-based methods (Boulch et al., 2017; Lawin et al., 2017; Wu et al., 2018; Milioto et al., 2019) and volumetric-based methods (Rethage et al.,2018; Meng et al., 2019; Boulch, 2020). Different from the above two kinds of methods, point-based methods, such as PointNet (Qi et al., 2016a), PointNet++ (Qi et al., 2017a), PointCNN (Li et al., 2018), and PointConv (Wu et al., 2019a), process 3D point clouds directly. Because of avoiding information loss in data conversion, point-based methods generally achieve better performance (Guo et al., 2021). Graph-based methods, such as DGCNN (Wang et al., 2019), KPConv (Thomas et al., 2019) can be considered as the extension of point-based methods. They utilize the graph as the representation and

* Corresponding author.
E-mail addresses: kent-zhipeng.luo@polyu.edu.hk, lswzshi@polyu.edu.hk (W. Shi), junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.jag.2022.102904
Received 14 February 2022; Received in revised form 20 May 2022; Accepted 29 June 2022
Available online 4 August 2022

obtain more larger receptive field than point-based methods. All these methods have been applied in indoor and outdoor datasets. Some of them achieve excellent performance on both accuracy and efficiency.

For most existing methods, they tend to stack directly different point properties, such as the coordinates and intensity, as the raw input. However, these properties are heterogeneous. More specifically, there are modality gaps between coordinates and intensity. The point coordinates, x, y, and z, describe the location information of point clouds in 3D space, while the point intensity records the laser reflection, which mainly captures the material properties of object surface. They can be considered as two uncoupled modalities. Therefore, simply stacking them as input will make the network tend to mine the modality-agnostic features, while ignore the valuable modality-specific information.

A feasible way to resolve this problem is processing these two modalities in a separate way to obtain the initial features, and designing an appropriate fusion network to embed them efficiently. In this paper, we first design an intensity-distribution aware (IDA) descriptor to mine the intensity distribution pattern. It takes the raw point coordinates and intensity as input, and outputs a multi-channel mask for each point to represent the intensity distribution information. Then, we propose a two-stage fusion strategy, i.e., in training stage, intensity-distribution feature is used to guide the model training via a designed loss, while in testing stage, the obtained multi-channel mask is considered as an extra confidence-score information to refine the initial predicted results. Through experiments, the two-stage strategy is proved to be helpful in improving the performance of semantic segmentation significantly. We evaluated the proposed method on two indoor corridor areas. Experimental results show the necessity of the separating processing way, as well as the effectiveness of our designed embedding strategy. There are two main contributions of this paper:

1) We propose an intensity-distribution aware (IDA) descriptor to mine the intensity information and generate an appropriate form that can be fused into the model. IDA descriptor takes the raw point clouds as input, and outputs a multi-channel mask for each point to describes the intensity-distribution pattern.

2) We propose a two-stage embedding strategy and investigate other two common fusing methods, i.e., early- and mid- fusing ways. Compared with common embedding strategies, the designed two-stage fusion method makes full use of the intensity-distribution information, which can be integrated efficiently into model and obtain better performance.

The rest of this paper is organized as follows: Section 2 reviews 3D point clouds semantic segmentation and multi-modality fusing related methods. Section 3 introduces the proposed method. Section 4 shows and analyses the experimental results. Sections 5 concludes the paper.

2. Related work

2.1. 3D point clouds semantic segmentation

Traditionally, 3D point clouds semantic segmentation is based on handcrafted-feature descriptors, such as FPFH (Rusu et al., 2010), RoPS (Guo et al., 2013), SHOT (Salti et al., 2014) and Tri-Spin-Image (Guo et al., 2015). However, the performance is usually far from satisfactory. Currently, a lot of DL-based methods have been undertaken and achieve excellent results. Generally, these methods can be divided into four categories: projection-, discretization-, point- and graph- based methods (Guo et al., 2021).

Projection-based methods firstly project raw 3D point clouds into 2D images. As a pioneering work (Lawin et al., 2017), authors first projected the 3D point clouds into 2D images in multiple views, following by a regular 2D CNN. Besides, Boulch et al., (2017) transformed the 3D point clouds into several partial snapshots from multiple views, then applied 2D CNN to obtain the semantic result. Jaritz et al., (2019) projected local point clouds onto virtual tangent planes, then a well-designed tangent convolution is used to process these planes. In addition, spherical representation is another 2D image form. Wu et al., (2018) projected point clouds onto a sphere to obtain a dense and regular representation, then used the SqueezeNet (Iandola et al., 2016) to extract the semantic features. Finally, conditional random field is utilized to refine the predict results. The improved version, SqueezeSegV2 (Wu et al., 2019b), is proposed with an unsupervised learning pipeline. Besides, in Milioto et al., (2019), range image is taken to represent the raw point clouds, and regular CNN is exploited to predict each pixel. Although these spherical-image based methods enjoy excellent performance, they still inevitably suffer from some issues, such as the information loss in discretization and occlusions.

Discretization-based methods convert 3D point cloud into discrete representation. It is straightforward to use the voxels to represent the irregular point clouds. As a classic method, Huang and You (2016) transform point clouds into voxels and utilized a fully 3D CNN to predict each voxel. Based on Huang et al., (2016), Tchapmi et al., (2017) proposed SEGCloud to improve the semantic results. By introducing a variational autoencoder architecture, Meng et al., (2019) used RBF to obtain a continuous representation, which would be more superior than the common binary occupancy one. Focusing on large-scale point clouds processing, Rethage et al., (2018) proposed the FCPN to divided the point clouds into different levels of geometric relations. Different from the above methods, to reduce the memory and computation consumption, Graham et al., (2018) used the indexing structure to design a submanifold sparse convolution method. Similar works include the MinkowskiNet (Choy et al., 2019), SPLATNet (Su et al., 2018) and LatticeNet (Rosu et al., 2020). Although these methods can obtain regular representation and remain the space structure, the information loss and high time-and-memory consumption are still unavoidable problems.

Point-based methods directly process the 3D point clouds. As the pioneering work, PointNet (Qi et al., 2017a) utilizes the global pooling and point-wise MLPs to obtain the semantic features for each point. Building on this work, a lot of methods have been proposed. As an improvement of PointNet, PointNet++ (Qi et al., 2017b) designs the neighboring feature pooling to obtain a larger local receptive field. Aiming to obtain orientation information and multi-scale features, Jiang et al., (2018) proposed PointSIFT, which is an individual module so that can be inserted into different methods. Besides, utilizing the attention mechanism aggregation (Vaswani et al., 2017) is also an effective idea to obtain descriptive features. In Zhao et al., (2019), authors designed an attention-based score refinement module to improve the segment performance. In Yang et al., (2019), a group shuffle attention is proposed to obtain more informative local domain relationships. In addition, several methods applied the point convolution to improve the effectiveness. As a representative work, PointCNN (Li et al., 2018) provides an x-transformation to learn the relations between local neighbor points. Similarly, in KPConv (Thomas et al., 2019), authors proposed a kernel point convolution, which would encode more neighborhood information and enjoy high robustness.

Graph-based methods can be considered as the extension of point-based methods. They use the graph structure to represent 3D point clouds. DGCNN (Wang et al., 2019) is one of representative graph-based works. It firstly constructs the graph directly from raw point clouds using Euclidean distance. Each point is considered as a node in the graph. Then, common MLPs is operated on the graph. Besides, in Landrieu and Simonovsky (2018), super-point graph is utilized to represent point clouds, and as the improvement version, in Engelmann et al., (2018), authors provided a deep metric learning model with a well-designed graph-structured contrastive loss. In addition, to obtain more descriptive geometric features, several methods have been provided, such as PyramNet (Kang and Li, 2019), PointGRC (Ma et al., 2020) and Randla-Net (Hu, et al., 2021).

2.2. Multi-modality fusion

Since multi-modal data usually enjoys highly nonlinear relations

between different modalities (Liu et al., 2018), mining information from this data efficiently is the key issue in many fields. For 3D point clouds processing, several fusion methods have been undertaken and achieve satisfactory results. Gonzalez et al., (2017) designed a mixture-of-experts model with fusing image, depth and optical flow information for pedestrian detection. Luo et al., (2019) proposed JointNet, which jointly learning information from multi-view and hand-created features for object recognition. In Hamraz et al., (2019), authors designed a multi-modality model to embed point intensity and other point attributes for deciduous classification. As for fusing with intensity, Wen et al., (2019) proposed a deep learning framework for 3D road marking processing with full use of the point intensity. Huang et al., (2020) provided a deep SAR-Net for SAR image processing by utilizing the intensity information. Zhu et al., (2021) fused the point intensity and aerial images for multi-modality registration. All these methods make full use of point attributes, such as intensity, depth, and coordinates, which brings the performance gain. However, these methods rarely explore more efficient fusion for different information, which leads to the low fusion efficiency. To obtain a higher modal fusion efficiency, by combing with attention strategy, Luo et al., (2020) provided RSSNet to fuse the slice- and point-level information for 3D object retravel and recognition. In addition, Xiao et al., (2021) also proposed FPS-Net by fusing point coordinates, intensity and depth in an effective multi-modality learning way.

3. The proposed method: IDA-Net

3.1. Over view

As presented in Fig. 1, the proposed method contains two branches. The up branch, IDA descriptor, is designed to mine intensity distribution information, while the down branch is used to extract the point-level features and obtain the initial predict results. In addition, a two-stage fusion strategy, i.e., guiding in training stage and refining in testing stage, is proposed to embed the intensity distribution information into model. We will introduce IDA descriptor and the two-stage embedding strategy in the following subsections.

3.2. IDA descriptor

As shown in Fig. 1, IDA descriptor consists of two main operations, the multi-peak searching (MPS) and intensity distribution mapping (IDM). MPS is designed to compute the multiply peaks from each histogram, while IDM contains three probability mappings, which take peaks as input, and outputs the intensity distribution (ID) mask for each point.

3.2.1. Multi-peak searching (MPS)

MPS contains three main steps, histogram constructing, peak searching, and appropriate peak selecting.

Histogram constructing. Given the point set $P = \{p_i \mid p_i = (x_i, y_i, z_i), i = 1, 2, \ldots, n\}$, MPS first generates the histogram along each axis by designing histogram generation function, H. We take $\times$ axis case as example. $H(x)$ first splits point set P into different subsets,

$$B = \{B_i | i = 1, 2, ..., k\}, B_i = \{p_j | p_j \in P, \lfloor x_j \times k \rfloor = i\}, \tag{1}$$

where B_i denotes the i^{th} subset, x_j is the $\times$ coordinate value of point p_j, $\lceil \rceil$means up rounding function, $k=\lceil L_x/r \rceil$ is the number of subsets, and L_x is the length of point set P along $\times$ axis, i.e.,

$$L_x = \max(\{x_j\}, j = 1, \cdots, n) - \min(\{x_j\}, j = 1, \cdots, n), \tag{2}$$

and r is the resolution. Then, these subsets are considered as the bins of histogram. We define the following three-element vector as the value of each bin,

$$W_{B_i} = \left(\frac{|B_i|}{|P|}, \text{mean}\left(\{I_{p_j}\}\right), \text{var}\left(\left\{I_{p_j}\right\}\right) \right), p_j \in B_i \tag{3}$$

where W_{B_i} is the value of i^{th} bin, $|\ |$ is the cardinal number of set,I_{p_j} is the intensity of point p_j. It is worth noting that the first element in W_{B_i} describe the spatial distribution of points, while the last two elements capture the intensity distribution. "*mean*" and "*var*" denote the mean and variance of point intensity in i^{th} bin. Points belong to the same category tend to have the same intensity, which is consistent with the real-world scene. Therefore, the second element is used to assist in determining which category these points belong to, while the third element can be used to determine whether the corresponding bin

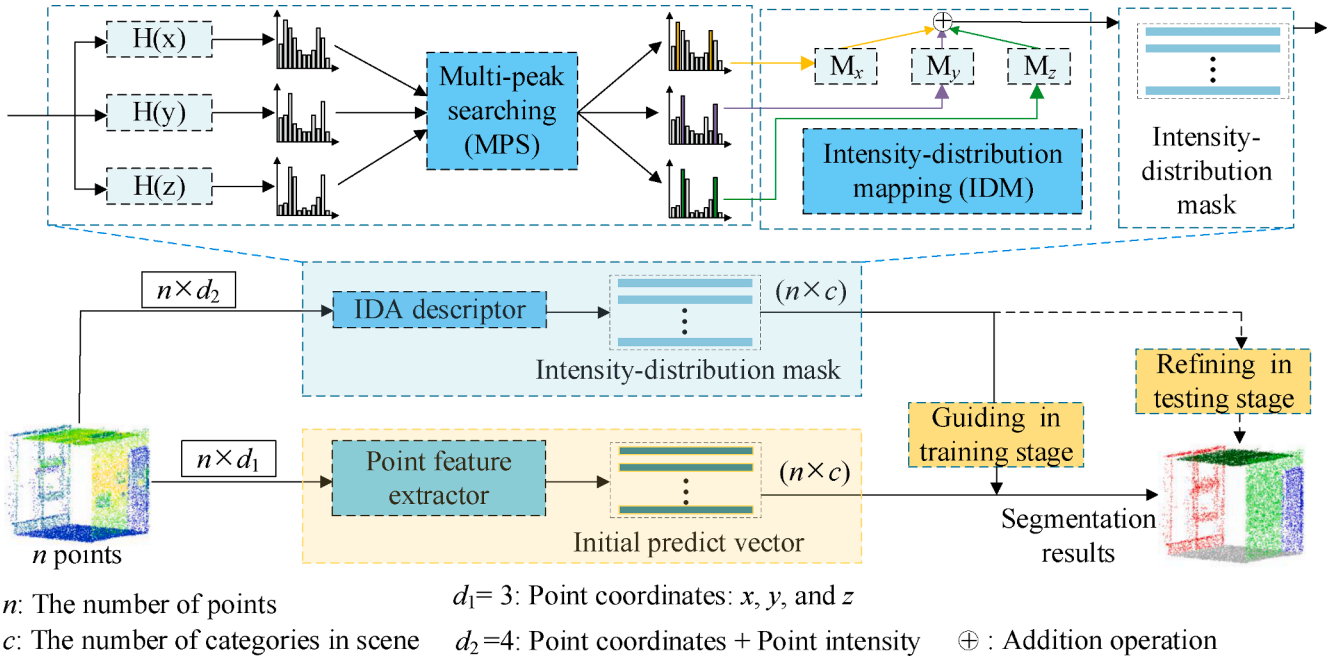


Fig. 1. Architecture of IDA-Net.

contains only a single category. By this way, we can obtain the histogram along x axis. Histograms along y and z axes can also be generated using the above method.

It needs to point that, r is an important parameter. Since the ranges of three axis are normalized to [0,1], r determines the width of bin. Generally, a smaller r means each bin enjoys a smaller width and so the corresponding histogram can capture more detail information. However, this also means higher computation consumption and lower robustness to noise. Section 4 will discuss the effect of r.

Peak searching. Generally, peaks with high bin values would provide more meaningful information. For convenience, we provide the following simple definition to describe the peak.

Definition 1. *Given a histogram and the set of bins* $B = \{B_i \mid i = 1, 2, \ldots, k\}$, $\forall\, B_i$, $i = 2, 3, \ldots, (k\text{-}1)$, *if* $W_{B_i}(1) \geq W_{B_{i+1}}(1)$ *and* $W_{B_i}(1) \geq W_{B_{i-1}}(1)$, *then* B_i *is a peak of the histogram. In addition, for* B_1 (B_k), *if* $W_{B_1}(1) \geq W_{B_2}(1)$ ($W_{B_k}(1) \geq W_{B_{k-1}}(1)$), *then* B_1 (B_k) *is also a peak of the histogram.* $W_{B_i}(1)$ *is the first element in the value vector of* B_i, *as defined in equation* (3).

According to definition 1, peak searching can be finished simply.

Appropriate peak selecting. Supposing all peaks have been obtained and arranged in descending order according to their bin values, i.e., and $B_{peak} = \{B_i' | W_{B_i'}(1) \geq W_{B_{i+1}'}(1), i = 1, 2, \cdots, m\}$, where m is the number of peaks. We utilize the bin value accumulation way to select the appropriate peaks. Similarly, we first provide a definition of the appropriate peaks as definition 2:

Definition 2. *Given a threshold* $\varepsilon \in (0,1]$ *and the set of peak bins with descending order, i.e.,*$B_{peak} = \{B_i' | W_{B_i'}(1) \geq W_{B_{i+1}'}(1), i = 1, 2, \cdots, m\}$, *if* $\exists\, l \in \{1, 2, \ldots, m\}$, *s.t.* $\sum_{i=1}^{l} W_{B_i'}(1) < \varepsilon$ *and* $\sum_{i=1}^{l+1} W_{B_i'}(1) \geq \varepsilon$, *then* $B_{appro-peak} = \{B_1', B_2', \cdots, B_l', B_{l+1}'\}$ *are the appropriate peak set.*

According to definition 2, appropriate peak selecting can be achieved simply by computing the bin value accumulation. Note that, since $W_{B_i'}(1) \in (0, 1), \forall i \in \{1, 2, \ldots, m\}$ and $\sum_{i=1}^{l+1} W_{B_i'}(1) < \sum_{i=1}^{n} W_{B_i'}(1) = 1$, there must exist l satisfying the conditions in definition 2 with given the threshold $\varepsilon \in (0,1]$. Built on the above three steps, we can run MPS operation and obtain the appropriate peaks for downstream steps.

3.2.2. Intensity distribution mapping (IDM)

As shown in Fig. 2, IDM contains two steps, interval determination and intensity distribution calculation.

Interval determination. Interval determination is achieved by a simple threshold strategy. As shown in Fig. 2, given a threshold τ, for each appropriate peak B_i', its neighbor bins can be obtained by comparing the bin values and the threshold. We provide algorithm 1 to describe the strategy.

Algorithm 1: Selecting the neighbor bins for given appropriate peak

Input: Appropriate peak B_i' Threshold τ; The set of bins $B = \{B_i \mid i = 1, 2, \ldots, k\}$

(*continued on next column*)

(*continued*)

Algorithm 1: Selecting the neighbor bins for given appropriate peak

Output: The set of neighbor bins $B_{neighbor}$

$B_{left} = [\,]$; $B_{right} = [\,]$;
$u = 1$; $v = 1$;
While $W_{B_{i-u}}(1) > \tau$ **do**
  $B_{left} \leftarrow B_{i-u}$; $u = u + 1$;
While $W_{B_{i+v}}(1) > \tau$ **do**
  $B_{right} \leftarrow B_{i+v}$; $v = v + 1$;
$B_{neighbor} \leftarrow [B_{left}, B_{right}]$
return: $B_{neighbor}$

Intensity distribution calculation. This operation is designed to obtain the ID mask. For each appropriate peak B_i', through the above steps we have obtained its neighbor bin set, denoted as $\{B_{i-u}, B_{i-u+1}, \cdots, B_i, B_{i+1}, \cdots, B_{i+v-1}, B_{i+v}\}$. Exponential function is adopted to describe the intensity distribution pattern. More specifically, for the appropriate peak B_i', its mean intensity is used to obtain the probability mask. The mean of point intensity for each category is calculated first by utilizing the training data, denoted as $\{mean_1, mean_2, \cdots, mean_c\}$, where c is the number of categories. Then the probability mask for each point in in B_i' is defined as,

$$\left(f\left(W_{B_i'}(2), mean_j\right)\right)_{j=1}^{c}, f(a,b) = e^{-|a-b|} \tag{4}$$

For each bin B_i in the neighbor bin set of B_i', its variance of point intensity is used,

$$f(W_{B_i}(3), 0) \bullet \left(f\left(W_{B_i'}(2), mean_j\right)\right)_{j=1}^{c} \tag{5}$$

Therefore, the intensity distribution mapping M can be written as,

$$M(p) = \begin{cases} \left(f\left(W_{B_i'}(2), mean_j\right)\right)_{j=1}^{c}, p \in B_i' \\ f(W_{B_i}(3), 0) \bullet \left(f\left(W_{B_i'}(2), mean_j\right)\right)_{j=1}^{c}, p \in neighborbinsetof B_i' \\ (0)_{j=1}^{c}, others \end{cases} \tag{6}$$

Appling M on histograms along $\times$, y and z axes, we can obtain M_x, M_y, and M_z respectively. A point-wise addition operation is utilized to generated the final ID mask, i.e.,

$$Mask(P) = M_x + M_y + M_z \tag{7}$$

3.3. Two-stage embedding strategy

Embedding extra information into model is the key problem in multimodal task. In this paper, we design a two-stage embedding strategy to fuse the intensity distribution information. Specifically, in training stage, we utilize ID information to construct a loss function, which is used to guide the model training. In testing stage, ID mask is used to

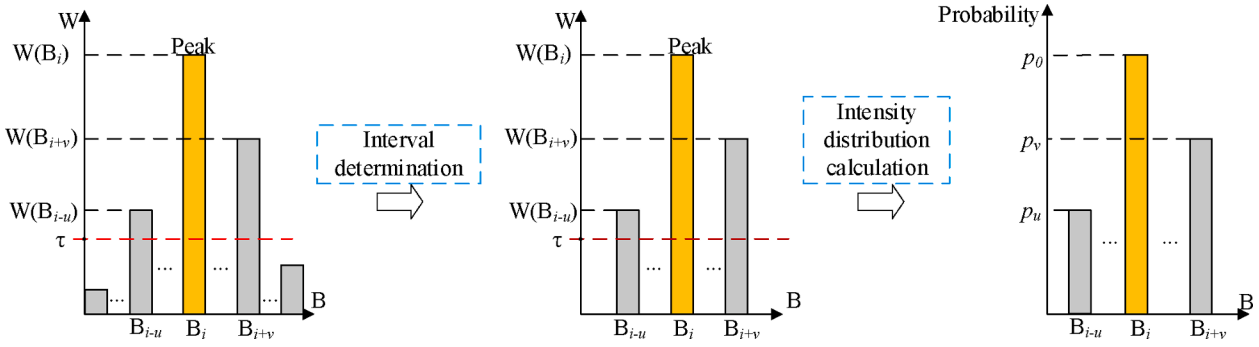


Fig. 2. Diagram of intensity distribution mapping (IDM).

refine the initial predict results in perspective of information compensation, to obtain the final prediction. In addition, we provide two basic embedding strategies as the comparison methods.

3.3.1. The proposed fusion strategy

As shown in Fig. 1, through IDA descriptor and point feature extractor, we obtain the ID mask, $V^{ID} = \{v_i^{ID} \mid i = 1, 2, \ldots, n\}$, where v_i^{ID} is a $1 \times c$ vector, and initial predicting result, $V^{ini} = \{v_i^{ini} \mid i = 1, 2, \ldots, n\}$, where v_i^{ini} is also a $1 \times c$ vector. Considered that ID mask contains important intensity distribution, it is meaningful to use it as a guidance signal. In this work, we propose a confidence filtering method to fuse intensity information. The core idea is to keep the prediction results with high confidence and abandon the low confident results. Specifically, as shown on Fig. 3, we first adopt a subsection mapping to filter the ID mask,

$$F_\alpha : R^{1\times c} \rightarrow R^{1\times c}, F_\alpha(v) = \begin{cases} v, \max(v) \geq \alpha \\ (0, 0, \cdots, 0)_{1\times c} others \end{cases} \quad (8)$$

where α is the confidence threshold. Then, we design the ID loss as,

$$loss_p = \frac{1}{N}\sum_N \left(\frac{1}{|\Lambda|}\sum_{j\in\Lambda}^{|\Lambda|}\left(v_j^{ID} - v_j^{ini}\right)^2\right) \quad (9)$$

where N is the batch size, Λ is the subscript set generated by the above subsection mapping, i.e.,

$$\Lambda = \{i | v_i \in V^{ID}, \max(v_i) \geq \alpha\} \quad (10)$$

In addition, the common loss function, i.e., cross entropy loss function, is used to evaluate the difference between initial predict result and ground truth,

$$loss = -\frac{1}{N}\sum_N Y\log(V^{ini}) \quad (11)$$

Therefore, the final loss for the model can be written as,

$$L = loss + \beta loss_p \quad (12)$$

where β is the weight of $loss_p$.

In testing stage, we take ID mask as the compensation information to refine the prediction. More specifically, ID mask V^{ID} is considered as a filter, which is operated on the initial predicting result V^{ini} to obtain the final fusion result V^f,

$$V^f = V^{ini} * \left(1 + V^{ID}\right) \quad (13)$$

where '*' means the element-wise multiplication. In this way, we embed the prior information into the result.

3.3.2. Two basic embedding strategies

In this section, build on early- and deep- fusion ways, we provide two basic embedding strategies as the comparison methods.

As shown in Fig. 4(a), embedding strategy I is an early-fusion method. The ID mask is merged directly into the raw point clouds P by a concatenation operation to obtain the finial input as follows,

$$V^{input} = \text{Concate}(P, V^{ID}). \quad (14)$$

Embedding strategy II is a deep-fusion method. More specially, ID mask V^{ID} is embedded into model in feature extraction stage. As shown in Fig. 4(b), for each convolution operation, ID mask is first merged with the output of previous operation, then the merged result is input into the convolution. The processing can be written as,

$$Output_{Convi} = \text{Conv}_i\ (\text{Concate}\ (P, Output_{Convi\text{-}1})). \quad (15)$$

These two embedding strategies are taken as the comparison methods. In the following experimental section, we will investigate the performance of each embedding strategy.

4. Experiments

Several experiments are conducted to evaluate the proposed method. Section 4.1 describes the dataset used in this paper. Section 4.2 presents the prior module validation. Section 4.3 analyzes the experimental results, and compares the proposed method with other two embedding strategies.

4.1. Dataset description and implementation

The datasets were collected by an MLS system on two indoor corridor areas, which are on different floors. As shown in Fig. 5, the length of corridor in area 1 and 2 is about 250 m and 300 m respectively. Area 1 contains five main categories, i.e., {Floor, Ceiling, Wall, Door, Window frame}, while area 2 does not contain window frames. About 120 m and 100 m areas are selected as training samples from areas 1 and 2, respectively. Therefore, in this work, there are about 220 m training area and 330 m testing area. The intensity information is utilized as one of point attributions. Besides, the mean of point intensity for each category is calculated by utilizing the training area, which is described in section 3.2.2. The result is $\{mean_1, mean_2, \cdots, mean_5\}$={8.84, 65.42, 67.07, 23.88, 22.16}.

Implementation. The model was trained on a Quadro GV100 GPU. Since the proposed model utilizes different architectures as backbones, the hyperparameters, such as the batch size, initial learning rate and number of epochs, follow that of the corresponding backbone. In addition, two factors descripted in section 3.3.1, the confidence threshold, α, and the weight of $loss_p$, β, are experimentally set to 0.8 and 0.1, respectively.

4.2. IDA descriptor validation

As discussed in section 3, three important parameters are taken in IDA descriptor, the resolution of bin r, the accumulation threshold ε, and the IDM operation threshold τ. In this section, we firstly analyze the above three hyper-parameters generation, then validate the performance of IDA descriptor.

4.2.1. Parameters generation

We use grid search method to obtain the optimal combination. The optional sets for r, ε, and τ are set to {0.003, 0.005, 0.007, 0.009, 0.012}, {50%, 55%, 60%, 65%, 70%,} and {0.005, 0.015, 0.025}, respectively. The following error rate is taken to evaluate the difference between the

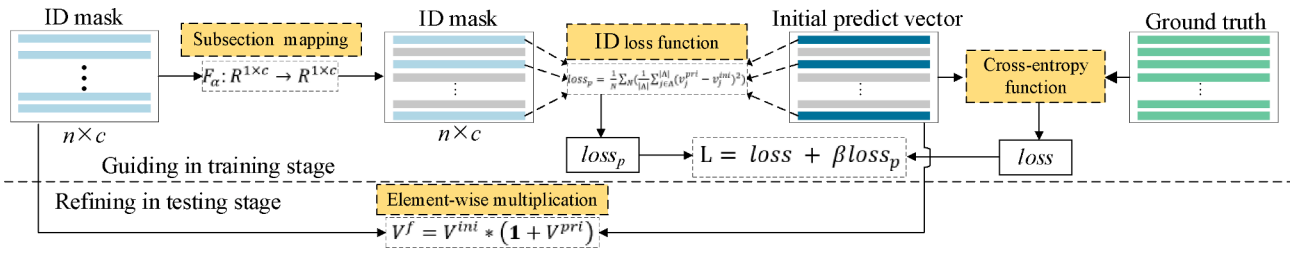


Fig. 3. Diagram of the proposed two-stage embedding strategy.

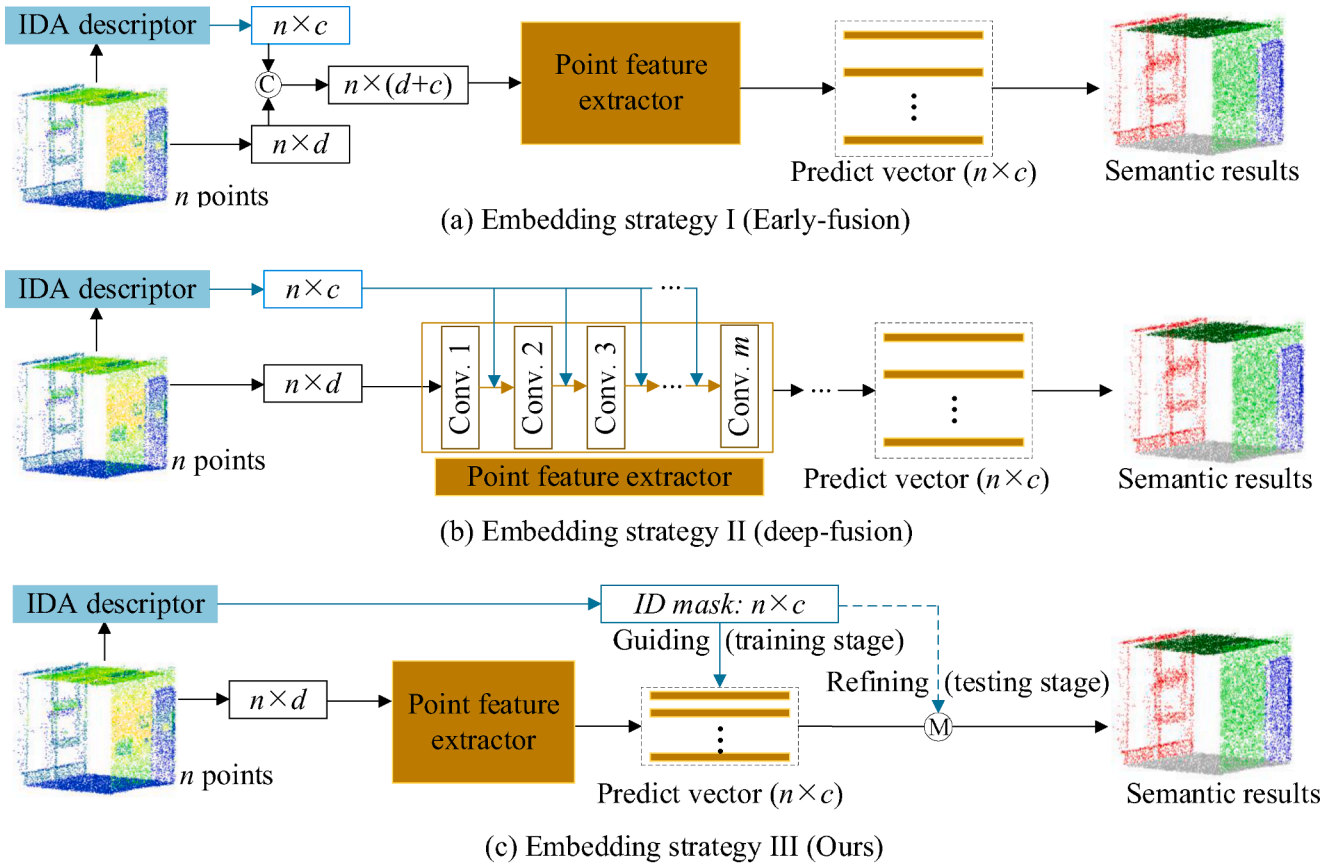


Fig. 4. Diagrams of three comparison embedding strategies.

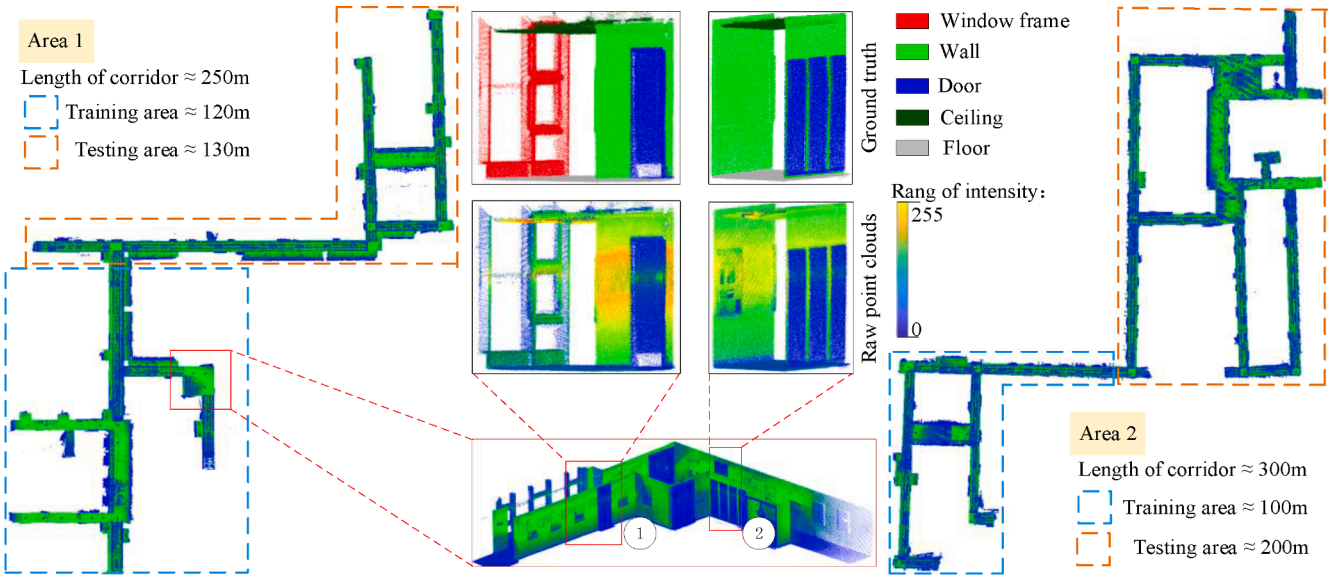


Fig. 5. Two indoor corridor areas and several samples with ground truth.

generated ID mask and ground truth:

$$Errorrate = \frac{1}{n}\sum_{i=1}^{n}(V_i^{ID} - V_i^{gt})^2 \quad (16)$$

where n is the point number, V^{ID} is the ID mask, and V_i^{gt} means the i^{th} element of ground truth. It is easy to see that the lower the error rate, the higher the similarity between the ID mask and ground truth, and so the better the performance of IDA descriptor. Note that, only training areas

are used in this parameter generation. By varying parameters, the error curves can be obtained, as shown in Fig. 6.

The resolution r determines the width of bin. A bin with a larger r can encode more spatial distribution information from the raw point clouds. However, when the resolution exceeds a certain threshold, the ability to describe the spatial distribution of bin will be weakened. As shown in Fig. 6, when the resolution increases from 0.003 to 0.005, the error rate decreases, which means the generated prior module enjoys a better performance. However, as r further increases, especially showing in Fig. 6(b), the error rate increases. Therefore, we set $r = 0.005$ as the optimal value.

The accumulation threshold ε effects the appropriate peaks searching. Similarly, a larger ε may help the searching algorithm to obtain more potential appropriate peaks, i.e., a higher recall. However, it will also deteriorate the precision, i.e., several inappropriate peaks may be selected. As shown in partial enlarged drawings of Fig. 6, especially Fig. 6(a) and (c), the error rate decreases as ε increases from 50% to 65%. However, as the above analysis, if ε further increases to 70%, the error increases. To obtain a better balance between recall and precision, we set the accumulation threshold as $\varepsilon = 65\%$.

The SDPM operation threshold τ is used to guide the generation of prior mask. Generally, a large τ means that the selection condition is strict, i.e., the value of the corresponding bin is higher, so that the results will be more accurate. However, a larger τ will also increase the number of missed selection points. As shown in Fig. 6, error rate with $\tau = 0.005$ and 0.015 is smaller than that with $\tau = 0.025$ when $r = 0.005$ and $\varepsilon = 65\%$. Considering that result with $\tau = 0.015$ is more stable than $\tau = 0.005$, we set $\tau = 0.015$.

4.2.2. IDA descriptor evaluation

Several samples are shown to demonstrate the processing of IDA descriptor. As presented in Fig. 7, given raw point clouds, histograms along x-axes are generated firstly. Then MPS operation is taken to search the appropriate peaks. Generally, about 2 or 3 appropriate peaks are searched. Finally, IDM operation is used to select neighbor bins and outputs the ID mask. From the results, we observe that the generated ID mask can capture the spatial distribution of points along x-axis. For example, for sample, there are three significant point clusters in raw point clouds along x-axis. After processing by MPS, three peaks and the corresponding neighbor bins are all detected.

In addition, as given in equation (3), the value of bin is defined as a three-element vector. We visualize these three elements to demonstrate the effectiveness of the definition. Fig. 8(a), (b) and (c) show results of histogram using the first, second and third element, respectively. As shown in Fig. 8(a), the 4-th and 93-th bins are the peaks, which contain wall points, while in Fig. 8(b), we can find that the values of 4-th and 93-th bins are closed to 67.07, which is the mean point intensity for walls (provided in section 4.1). This demonstrate that points in these two bins tend to belong wall. Besides, as shown in Fig. 8(c), it can be observed that the values of 4-th and 93-th bins are very small, which means the variance of point intensity for these two bins is small. This is because points in these two bins tend to belong to the same category. These visualization results shows the rationality and effectiveness of the definition in equation (3).

4.3. IDA-Net validation

Several experiments are taken to validate the performance of the proposed. We take three comparing methods: (1) the plain baseline without any fusion strategy, i.e., the backbone (as shown in the point feature extractor part in Fig. 4) without using ID mask; (2) the backbone fused with ID mask via prior embedding strategy I, denoted as ESF (i.e., Early-Stage-Fusion); (3) the backbone fused with ID mask via with prior embedding strategy II, denoted as MSF (i.e., Multi-Stage-Fusion). Four classic point clouds segmentation architectures, PointNet, PointNet++, PointConv and KPConv, are used as backbones. In addition, IoU (Intersection over Union) and OAcc (Overall Accuracy) are taken as the metrics. We first present the numerical results for the quantitative analysis, then several visualization results are shown for the qualitative discussion.

4.3.1. Quantitative analysis

Tables 1 and 2 show the comparison results with different backbones and embedding strategies on two testing areas. Overall, we can observe that IDA-Net obtains the excellent performance on each testing area. Specifically, on testing area 1, comparing with baseline, IDA-Net achieves a significant improvement in terms of mIoU (mean IoU) of 14.14%, 11.45%, 7.46% and 9.82% with four backbones respectively. On testing area 2, the gain is 19.91%, 10.53%, 25.64% and 7.36% respectively. These results indicate that fusing with intensity information can bring improvement for existing semantic segmentation models.

For different fusion strategies, it can be observed form Tables 1 and 2 that our method achieves the best performance. Comparing with MSF (i.e., embedding strategy II), IDA-Net obtains a gain of 3.66%, 2.53%, 3.57% and 1.72% in terms of mIoU with four backbones on testing area 1 respectively, while the improvement is 12.10%, 6.86%, 4.11% and 6.24% on testing area 2. These results show the superiority of the designed two-stage embedding strategy. The reason mainly lies in that the intensity and point coordinates are heterogeneous. This means simply stacking them as input, just as the way in MSF and ESF strategies, will make the network tend to mine the modality-agnostic features, but ignore the valuable modality-specific information. By contrary, the proposed strategy processes these two different modalities respectively and embedding them in a two-stage way, which would improve the utilization efficiently. In addition, we can observe that there is small difference between the performance of MSF and ESF. This means that no matter fusing the ID mask in early- or multi-stage way, the model has the almost same level of utilization of intensity information.

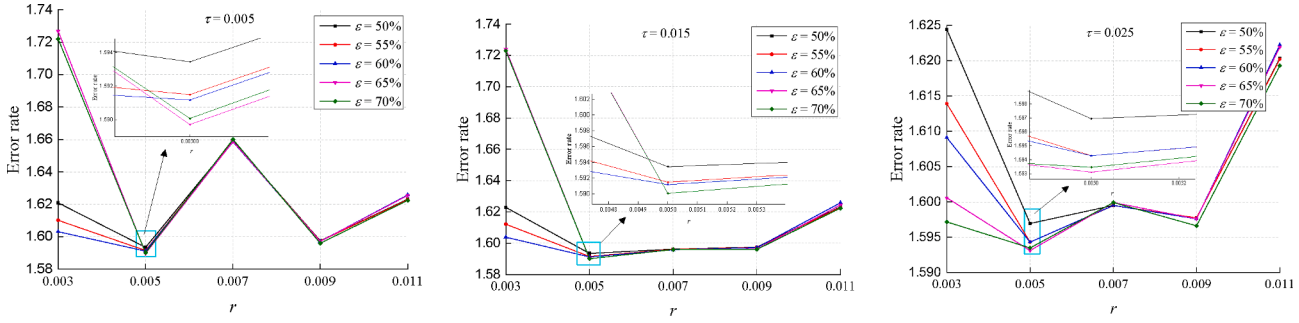


Fig. 6. Error curves generated by IDA descriptor with different parameter settings.

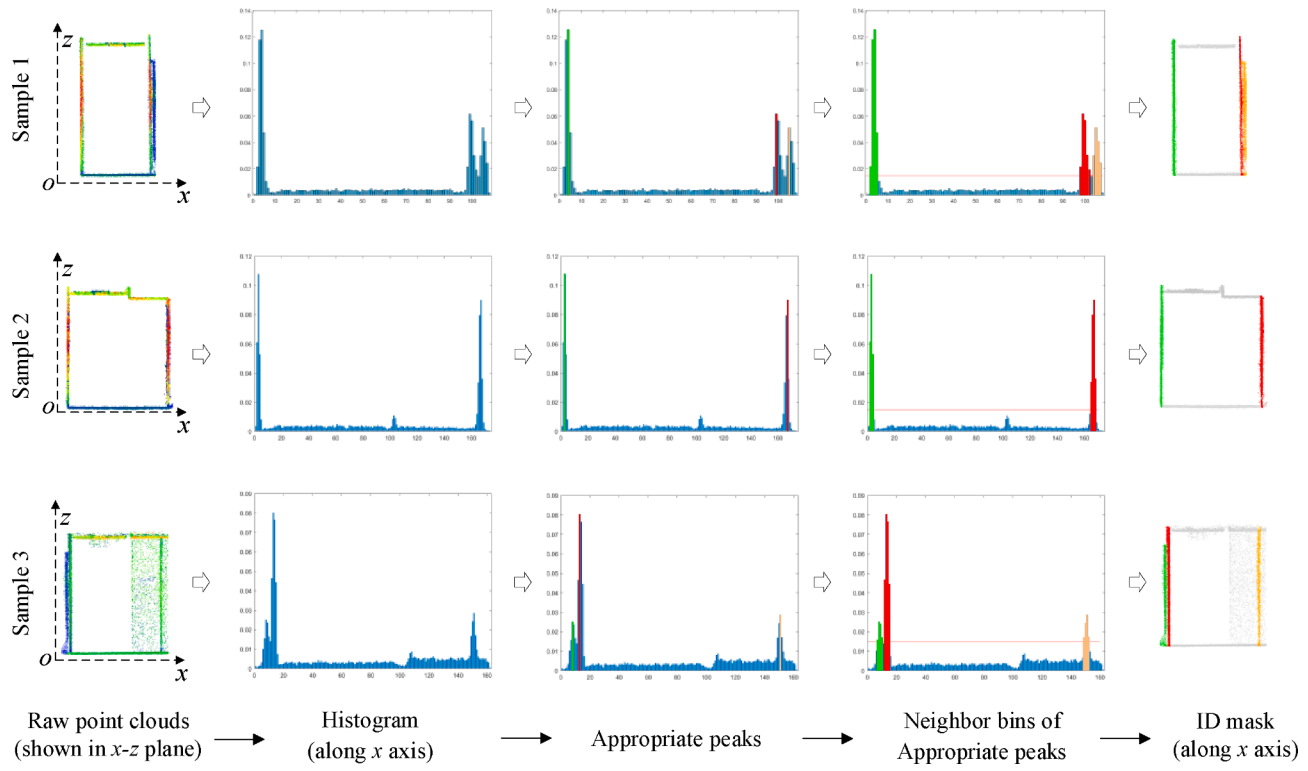


Fig. 7. Several samples taken to show the generation of ID mask.

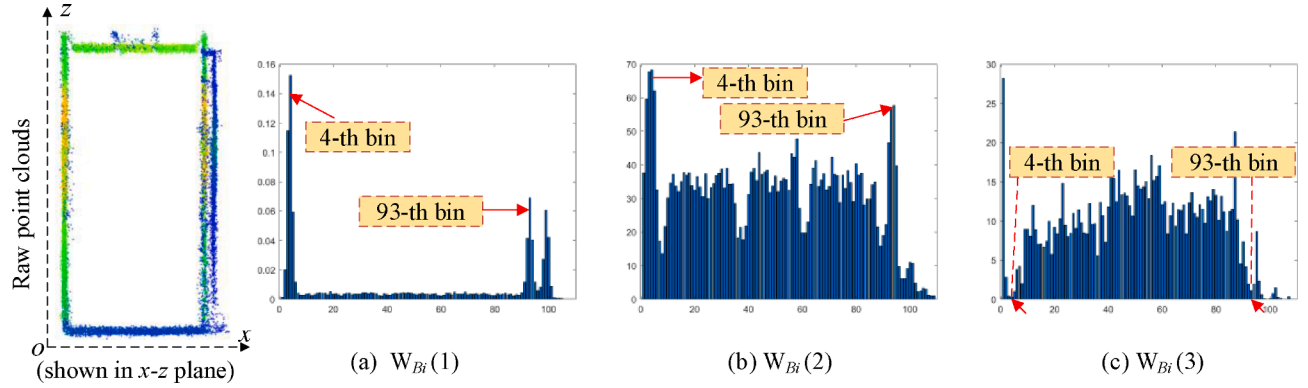


Fig. 8. Histograms using (a) the first element of bin value, (b) the second element (i.e., the mean of point intensity), and (c) the third element (i.e., the variance of point intensity).

Table 1
Comparison of different methods on testing area 1 with different backbones.

Backbone	Method	Floor	Ceiling	Wall	Door	Window frame	mIoU	OAcc
PointNet	Baseline	78.35	63.64	57.38	11.81	42.01	50.64	73.55
	ESF	92.51	69.94	66.76	22.56	51.89	60.73	80.42
	MSF	89.83	66.97	64.67	26.97	57.16	61.12	79.55
	Ours	85.49	76.69	69.70	30.43	61.59	**64.78**	**82.97**
PointNet++	Baseline	90.79	68.52	56.99	14.78	49.80	56.18	75.37
	ESF	95.32	64.69	71.55	50.47	43.10	65.03	83.33
	MSF	95.42	65.65	72.06	50.91	41.44	65.10	83.05
	Ours	97.08	79.38	68.29	32.59	60.81	**67.63**	**83.68**
PointConv	Baseline	96.97	96.44	74.22	26.40	76.74	74.15	82.98
	ESF	96.98	94.63	88.44	61.13	45.20	77.28	92.61
	MSF	96.72	95.24	88.58	62.55	47.13	78.04	**92.62**
	Ours	98.87	96.20	84.00	46.32	82.66	**81.61**	92.32
KPConv	Baseline	97.08	91.70	89.10	67.90	58.01	80.76	93.34
	ESF	97.30	94.76	93.44	86.12	71.86	88.70	96.25
	MSF	97.32	94.89	93.47	85.89	72.21	88.76	96.28
	Ours	98.39	95.97	94.58	85.73	78.23	**90.58**	**96.99**

Table 2
Comparison of different methods on testing area 2 with different backbones.

Backbone	Method	Floor	Ceiling	Wall	Door	mIoU	OAcc
PointNet	Baseline	92.22	50.91	10.48	8.62	40.56	36.17
	ESF	82.69	62.27	47.93	2.76	48.91	55.92
	MSF	89.74	54.78	40.52	8.44	48.37	51.39
	Ours	90.51	66.12	61.03	24.24	**60.47**	**65.81**
PointNet++	Baseline	88.96	66.97	56.95	1.67	53.64	68.04
	ESF	89.16	59.77	60.23	18.05	56.80	60.71
	MSF	88.84	62.58	62.79	15.03	57.31	61.63
	Ours	93.42	69.69	71.05	22.52	**64.17**	**69.17**
PointConv	Baseline	94.59	96.72	60.71	7.05	64.76	71.62
	ESF	95.49	96.08	83.55	30.82	76.48	78.25
	MSF	95.71	97.11	81.86	30.47	76.29	79.09
	Ours	95.70	97.67	87.41	40.83	**80.40**	**82.02**
KPConv	Baseline	95.09	95.34	80.03	55.87	81.58	87.50
	ESF	94.98	95.54	80.17	56.27	81.74	87.62
	MSF	95.50	96.58	80.42	58.28	82.70	88.06
	Ours	89.34	98.48	90.03	77.92	**88.94**	**93.91**

4.3.2. *Quantitative analysis*

Fig. 9 shows several semantic results with backbone of PointNet. Overall, it can be observed that our method outperforms other methods. Especially, our method is better for the semantic prediction of local structures, i.e., most of the points belonging to the same type of structure can be predicted correctly. For example, in samples 2 and 3, IDA-Net obtains excellent predicted result for the wall and door parts, while for other three comparing methods, their prediction results for the wall and door structures have several significant error points, as shown in areas marked by pink ellipses in Fig. 9. In addition, as shown in sample 1, for the ceiling part, especially the edges of ceilings and walls, our method also achieves the best performance.

These results can be explained from two aspects. On one hand, for ESF and MSF, they directly fuse the intensity information, which would cause a relatively low utilization. On the other hand, for the proposed method, the designed two-stage embedding mechanism, i.e., guiding the model training via a designed loss and refining the initial predicted results, will make the model achieve a better balance between features learning from coordinates and the intensity distribution information extracting from IDA descriptor.

In addition, there are also some mislabeled points in our method. As shown in sample 3, the area marked by a block rectangular box contains the mislabeled points. Points belonging to the window frame are incorrectly predicted as ceilings or walls. This may be because in the generation of ID mask, there are some deviations in the peak calculation process, which makes the generated ID mask inaccurate, and finally leads to some wrong prediction results. This suggests that we should explore more effective and robust mask generation methods in future work.

To obtain a better understanding for the performance of our method, we present the visualization on a larger-scale scene, as shown in Fig. 10. From Fig. 10 (a) and (b), it can be observed that our method obtains excellent results on ceiling and door. Almost all ceiling and door points can be rightly distinguished. However, as for the wall points, the left-most wall in Fig. 10 (a) and (b) is recognized as a door. This because in the corridor environment, the wall and door are almost inlaid with each other, which means they are easier to be misclassified. This case will be considered in our future work. From Fig. 10 (c) and (d), we can find that although some window frame points are mislabeled as the wall or the ceiling, the whole semantic segmentation results are acceptable. Overall, our method enjoys better performance in the semantic prediction of regular structure, especially the plane structure. This is consistent with the basic idea of this paper, i.e., regular structure can be easier mined by the designed ID descriptor and be embedded into the model, which makes the model achieve higher accuracy in the regular structure.

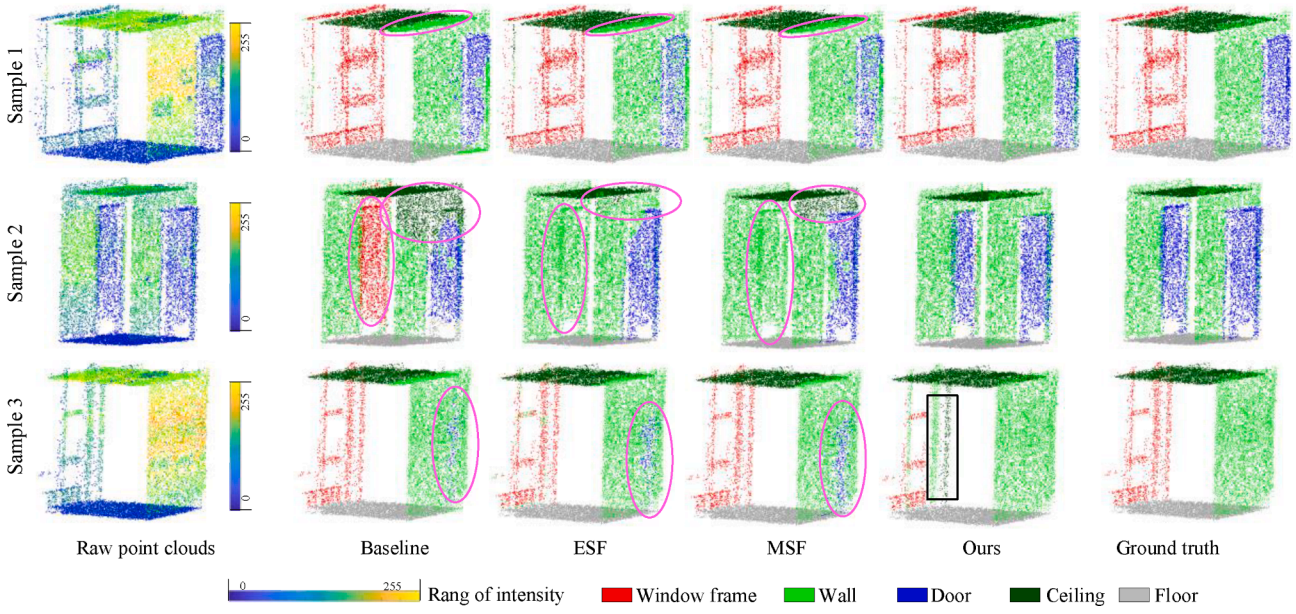


Fig. 9. Visualization of semantic results by different methods on several local scene.

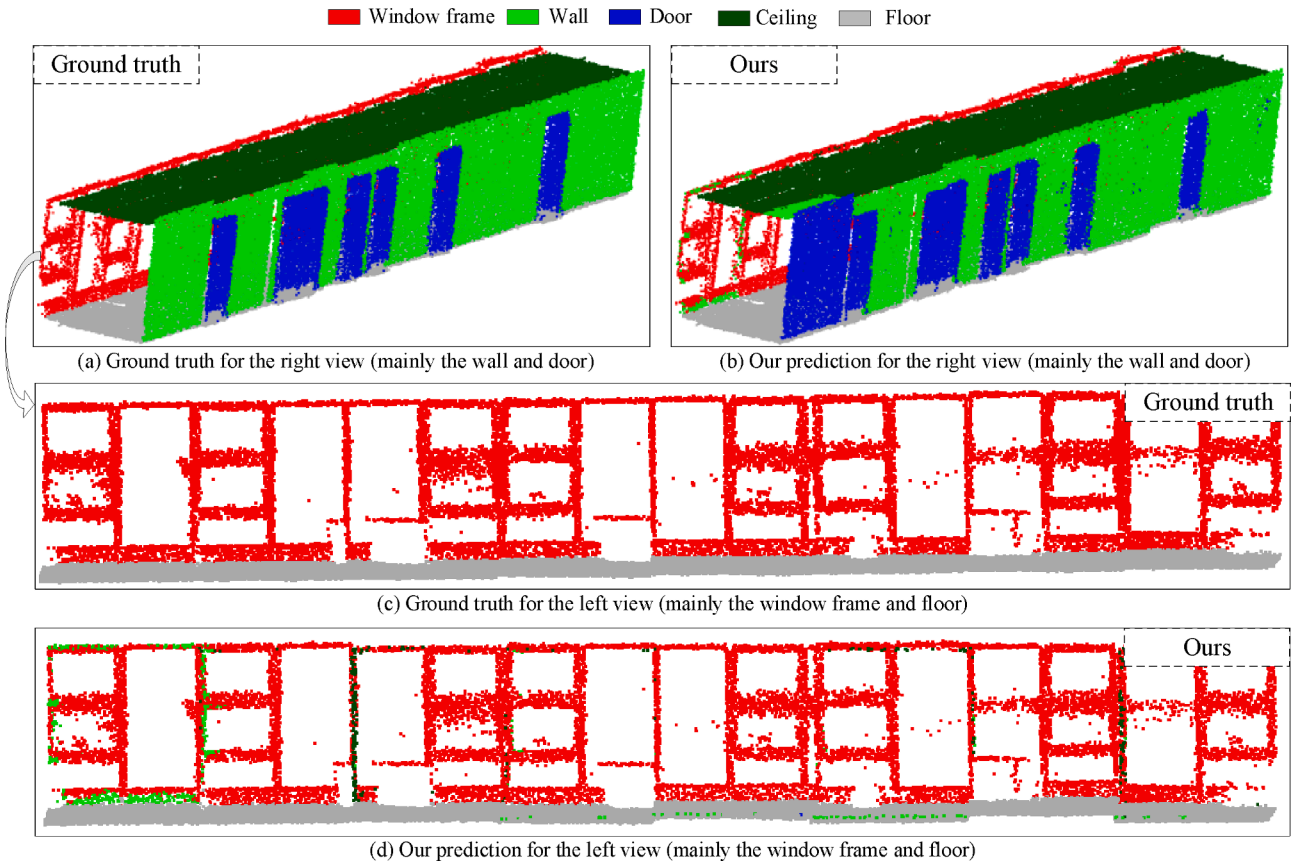


Fig. 10. Visualization of semantic results on a large-scale scene.

4.3.3. *Efficiency of IDA descriptor*

IDA descriptor is designed to obtain the ID mask. It can be divided into two steps: peak searching and ID mask generation. According to the analysis in section 3.2, the computation complexity of IDA descriptor is approximately linear, O(n), where n is the number of points. To further explore the time consumption of IDA descriptor, several experiments were conducted. More specifically, for a given sample, the point number range is set from 2048 to 16384. For each point number value, we run 1000 times and compute the average as the result. The time for each step is recorded. Note that for each sample, in addition to calculating peaks along the z-axis, the peak along the x/y axis is also calculated.

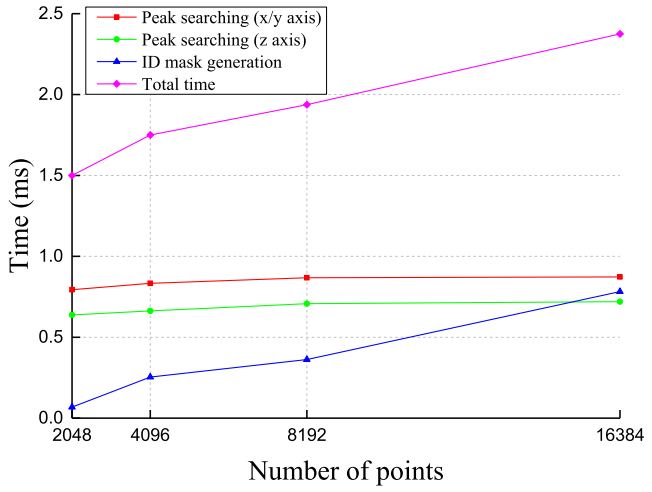


Fig. 11. Time consumption curves of IDA descriptor under different numbers of points.

Fig. 11 shows the time consumption curves under different numbers of points. Firstly, as shown by the pink curve, the total time consumption increases approximately linearly as number of points increases. This is consistent with the time complexity of O(n). Secondly, as shown by the red and blue curves, the time consumptions of peak searching along x/y and z axes remain almost constant as the number of points increases. This is because the time consumption of peak searching is only related to the number of bins, which only depends on the size (i.e., the length, width and height) of the sample size, independent of the number of points. Thirdly, as number of points increases, the time consumption of prior mask generation increases approximately linearly. This is because in prior mask generation, each point must be processed, which means the computation complexity is O(n). In summary, the time complexity of peak searching can be considered as a constant, while the total calculation time, less than 2.5 ms for each sample with 16,483 points, is acceptable in practice.

5. Conclusions

In this paper, we propose the IDA-Net, which contains an IDA descriptor and a well-designed two-stage fusion strategy, for the semantic segmentation of 3D indoor corridor. The above experimental results demonstrate the feasibility and necessity of the basic idea in this work, i.e., using the point intensity information can bring improvements for the model, as well as the superior performance of the proposed two-stage fusion strategy. These results can be summarized from two aspects. On one hand, for the common fusion ways, such as the ESF and MSF, they directly fuse the intensity information, which would cause a relatively low utilization. On the other hand, for the proposed two-stage fusion strategy, is will make the model achieve a better balance between leaning point spatial distribution features and mining the intensity distribution information.

Limitations and future work. The proposed IDA descriptor is suitable for scenes with clear spatial distribution, such as the indoor corridor environments. For scenes with large spatial distribution uncertainty, such as the road environments, the performance of our method may decrease. In the future work, we intend to investigate more robust and universal IDA descriptor, such as mining intensity distribution information in automatic data-driven ways, rather than the manual design ideas.

CRediT authorship contribution statement

Zhipeng Luo: Conceptualization, Methodology, Software, Writing – original draft. **Pengxin Chen:** Data curation, Writing – review & editing. **Wenzhong Shi:** Supervision. **Jonathan Li:** Validation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Otto Poon Charitable Foundation Smart Cities Research Institute, Hong Kong Polytechnic University (Work Program: CD03); Hong Kong Polytechnic University (1-ZVN6, 4-BCF7); The State Bureau Surveying and Mapping, P.R. China (1-ZVE8); and Hong Kong Research Grants Council (T22-505/19-N).

References

Boulch, A., 2020. ConvPoint: Continuous convolutions for point cloud processing. Comput. Graph. 88, 24–34.
Boulch, A., Saux, B.L., Audebert, N., 2017. Unstructured point cloud semantic labeling using deep segmentation networks. In: ACM Workshop 3D Object Retrieval. ACM, Lyon, France, pp. 17–24.
Broggi, A., Buzzoni, M., Debattisti, S., Grisleri, P., Laghi, M.C., Medici, P., Versari, P., 2013. Extensive tests of autonomous driving technologies. IEEE Trans. Intell. Transp. Syst. 14 (3), 1403–1415.
Choy, C., Gwak, J., Savarese, S., 2019. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, Long Beach, CA, USA, pp. 3070–3079.
Engelmann, F., Kontogianni, T., Schult, J., Leibe, B., 2018. Know what your neighbors do: 3D semantic segmentation of point clouds". In: Springer European Conference on Computer Vision Workshops (ECCVW). Springer, Munich, Germany, pp. 395–409.
Gonzalez, A., Vazquez, D., Lopez, A., Amores, J., 2017. On-board object detection: multicue, multimodal, and multiview random forest of local experts. IEEE Trans. Cybern. 47 (11), 3980–3990.
Graham, B., Engelcke, M., Maaten, L., 2018. 3D semantic segmentation with submanifold sparse convolutional networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, Salt Lake City, UT, USA, pp. 9224–9232.
Guo, Y., Sohel, F., Bennamoun, M., Lu, M., Wan, J., 2013. Rotational projection statistics for 3D local surface description and object recognition. Int. J. Comput. Vision 105 (1), 63–86.
Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., 2014. 3D object recognition in cluttered scenes with local surface features: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 36 (11), 2270–2287.
Guo, Y., Sohel, F., Bennamoun, M., Wan, J., Lu, M., 2015. A novel local surface feature for 3D object recognition under clutter and occlusion. Inf. Sci. 293, 196–213.
Guo, Y., Wang, H., Hu, Q., Liu, H., Bennamoun, M., 2021. Deep Learning for 3D Point Clouds: A Survey. IEEE Trans. Pattern Anal. Mach. Intell. 43 (12), 4338–4364.
Hamraz, H., Jacobs, N.B., Contreras, M.A., Clark, C.H., 2019. Deep learning for conifer/deciduous classification of airborne LiDAR 3D point clouds representing individual trees. ISPRS J. Photogramm. Remote Sens. 158, 219–230.
Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2021. Learning Semantic Segmentation of Large-Scale Point Clouds with Random Sampling. IEEE Trans. Pattern Anal. Mach. Intell. https://doi.org/10.1109/TPAMI.2021.3083288.
Huang, Z., Datcu, M., Pan, Z., Lei, B., 2020. Deep SAR-Net: Learning objects from signals. ISPRS J. Photogramm. Remote Sens. 161, 179–193.
Huang, J., You, S., 2016. Point cloud labeling using 3D convolutional neural network. In: IEEE International Conference on Pattern Recognition (ICPR). IEEE, Cancun, Mexico, pp. 2670–2675.
Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 MB model size. In: International Conference on Learning Representations (ICLR).
Jaritz, M., Gu, J., Su, H., 2019. Multi-view PointNet for 3D scene understanding. In: IEEE International Conference on Computer Vision Workshop (ICCVW). IEEE, Seoul, Korea (South), pp. 3995–4003.
Jiang, M., Wu, Y., Lu, C., 2018. PointSIFT: A sift-like network module for 3D point cloud semantic segmentation," arXiv preprint, arXiv:1807.00652.
Kang, Z., Li, N., 2019. PyramNet: Point cloud pyramid attention network and graph embedding module for classification and segmentation. In: Springer International Conference on Neural Information Processing (ICONIP). Springer, Sydney, NSW, Australia, pp. 35–43.
Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, Salt Lake City, UT, USA, pp. 4558–4567.
Lawin, F.J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F.S., Felsberg, M., 2017. Deep projective 3D semantic segmentation. In: Springer International Conference on Computer Analysis of Images and Patterns. Springer, Ystad, Sweden, pp. 95–107.
Li, Y., Bu, Rui., Sun, M., Chen, B., 2018. PointCNN: Convolution On X-Transformed Points. In: Annual Conference on Neural Information Processing Systems. Montréal Canada, pp. 828-838.
Liu, K., Li, Y., Xu, N., Natarajan, P., 2018. Learn to combine modalities in multimodal deep learning, arXiv preprint arXiv:1805.11730.
Luo, Z., Li, J., Xiao, Z., Mou, G., Cai, X., Wang, C., 2019. Learning high-level features by fusing multi-view representation of MLS point clouds for 3D object recognition in road environments. ISPRS J. Photogramm. Remote Sens. 150, 44–58.
Luo, Z., Di, L., Li, J., Chen, Y., Xiao, Z., Junior, J.M., Goncalves, W.N., Wang, C., 2020. Learning sequential slice representation with an attention-embedding network for 3D shape recognition and retrieval in MLS point clouds. ISPRS J. Photogramm. Remote Sens. 161, 147–163.
Ma, Y., Guo, Y., Liu, H., Lei, Y., Wen, G., 2020. Global context reasoning for semantic segmentation of 3D point clouds. In: IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, Snowmass, USA, pp. 2920–2929.
Meng, H.Y., Gao, L., Lai, Y.K., Manocha, D., 2019. VV-Net: Voxel vae net with group convolutions for point cloud segmentation. In: IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, Seoul, Korea (South), pp. 8499–8507.
Milioto, A., Vizzo, I., Behley, J., Stachniss, C., 2019. RangeNet++: Fast and accurate LiDAR semantic segmentation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Macao, China, pp. 4213–4220.
Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Las Vegas, Nevada, pp. 77–85.
Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Annual Conference on Neural Information Processing Systems. Long Beach, CA, USA, pp. 5105–5114.
Rethage, D., Wald, J., Sturm, J., Navab, N., Tombari, F., 2018. Fully convolutional point networks for large-scale point clouds. In: Springer European Conference on Computer Vision. Springer, Munich, Germany, pp. 625–640.
Rosu, R. A., Peer, S., Jan, Q., Sven, B., 2020. LatticeNet: Fast point cloud segmentation using permutohedral lattices. In: Robotics: Science and Systems (RSS). arXiv: 1912.05905.
Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J., 2010. Fast 3D recognition and pose using the Viewpoint Feature Histogram. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, Taipei, China, pp. 2155–2162.
Salti, S., Tombari, F., Stefano, L., 2014. SHOT: Unique signatures of histograms for surface and texture description. Comput. Vis. Image Underst. 125, 251–264.
Schreiber, M., Knppel, C., Franke, U., 2013. LaneLoc: Lane marking based localization using highly accurate maps. In: IEEE Intelligent Vehicles Symposium. IEEE, Gold Coast City, Australia, pp. 449–454.
Seo, Y., Lee, W., Zhang, J., Wettergreen, W., 2015. Recognition of highway workzones for reliable autonomous driving. IEEE Trans. Intell. Transp. Syst. 16 (2), 708–718.
Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.H., Kautz, J., 2018. SplatNet: Sparse lattice networks for point cloud processing". In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, Salt Lake City, UT, USA, pp. 2530–2539.
Tchapmi, L., Choy, C., Armeni, I., Gwak, J., Savarese, S., 2017. SEGCloud: Semantic segmentation of 3D point clouds. In: IEEE International Conference on 3D Vision (3DV). IEEE, Qingdao, China, pp. 537–547.
Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. KPConv: Flexible and deformable convolution for point clouds. In: IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, Seoul, Korea (South), pp. 6410–6419.
Uy, M.A., Lee, G.H., 2018. PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT, USA, pp. 4470–4479.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Annual Conference on Neural Information Processing Systems. Long Beach, CA, USA, pp. 6000–6010.
Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M., Solomon, J., 2019. Dynamic graph CNN for learning on point clouds. ACM Trans. Graphics. https://doi.org/10.1145/3326362.
Wang, W., Yu, R., Huang, Q., Neumann, U., 2018. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT, USA, pp. 2569–2578.

Wen, C., Sun, X., Li, J., Wang, C., Guo, Y., Habib, A., 2019. A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 147, 178–192.

Wu, B., Wan, A., Yue, X., Keutzer, K., 2018. SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. In: International Conference on Robotics and Automation. Brisbane, Australia, 2018, pp. 1887–1893.

Wu, W., Qi, Z., Li, F., 2019a. PointConv: Deep Convolutional Networks on 3D Point Clouds. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Montréal Canada, pp. 9613–9622.

Wu, S., Tateno, K., Navab, N., Tombari, F., 2020. SCFusion: Real-time Incremental Scene Reconstruction with Semantic Completion. In: International Conference on 3D Vision (3DV). Fukuoka, Japan, pp. 801–810.

Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K., 2019b. SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In: International Conference on Robotics and Automation (ICRA). Montreal, Canada, pp. 4376–4382.

Xiao, A., Yang, X., Lu, S., Guan, D., Huang, J., 2021. Robust registration of aerial images and LiDAR data using spatial constraints and Gabor structural features. ISPRS J. Photogramm. Remote Sens. 176, 237–249.

Yang, J., Zhang, Q., Ni, B., Li, L., Liu, J., Zhou, M., Tian, Q., 2019. Modeling point clouds with self-attention and gumbel subset sampling. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA, pp. 3318–3327.

Zhao, H., Jiang, L., Fu, C.W., Jia, J., 2019. PointWeb: Enhancing local neighborhood features for point cloud processing. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, Long Beach, CA, USA, pp. 5560–5568.

Zhu, B., Ye, Y., Zhou, L., Li, Z., Yin, G., 2021. Robust registration of aerial images and LiDAR data using spatial constraints and Gabor structural features. ISPRS J. Photogramm. Remote Sens. 181, 129–147.