TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 19/22 pp232–243
DOI: 10.26599/TST.2023.9010002
Volume 29, Number 1, February 2024

Dynamic Scene Graph Generation of Point Clouds with Structural Representation Learning

Chao Qi, Jianqin Yin*, Zhicheng Zhang, and Jin Tang

Abstract: Scene graphs of point clouds help to understand object-level relationships in the 3D space. Most graph generation methods work on 2D structured data, which cannot be used for the 3D unstructured point cloud data. Existing point-cloud-based methods generate the scene graph with an additional graph structure that needs labor-intensive manual annotation. To address these problems, we explore a method to convert the point clouds into structured data and generate graphs without given structures. Specifically, we cluster points with similar augmented features into groups and establish their relationships, resulting in an initial structural representation of the point cloud. Besides, we propose a Dynamic Graph Generation Network (DGGN) to judge the semantic labels of targets of different granularity. It dynamically splits and merges point groups, resulting in a scene graph with high precision. Experiments show that our methods outperform other baseline methods. They output reliable graphs describing the object-level relationships without additional manual labeled data.

Key words: scene graph generation; structural representation; point cloud

1 Introduction

Scene graph generation of the point cloud aims to recognize the objects and their contextual relationships from massive 3D points, which can be used in different tasks such as object retrieval in a 3D scene[1]. This task's core is extracting the structural information with semantics from the 3D point cloud. However, the point cloud is unstructured and contains millions or even tens of millions of discrete points[2]. It poses challenges for extracting structured relationships used for scene graph generations.

Scene graph generations have been widely explored in image-based tasks[3–5]. These works detect objects and establish relationships by analyzing patterns in the images. Unlike the images, which are regular data representing 2D scenes, the point clouds are irregular data representing 3D scenes. Thus, the image-based approaches cannot be directly used in the graph generation of point-cloud-based scenes.

In recent years, scene graph generations from 3D point clouds have been studied. These works focus on learning the semantic labels of nodes and edges on a given class-agnostic graph structure of the point cloud[6, 7]. Specifically, a manually annotated graph records the points-to-node mapping relationships and the connection states between nodes (with or without connection). This given graph structure promotes the effectiveness of the final graph generation. However, manual annotation is labor-intensive and strongly dependent on experience. Thus, it is urgent to explore an automatic way to extract the class-agnostic graph structure from the original point cloud.

In summary, the problems of current graph generation methods motivate us to explore a way to achieve graph generation from original point clouds without the help

• Chao Qi is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Standard and Metrology Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China. E-mail: qichao@bupt.edu.cn.
• Jianqin Yin, Zhicheng Zhang, and Jin Tang are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: {jqyin, zczhang, tangjin}@bupt.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2022-08-25; revised: 2022-11-09; accepted: 2023-01-06

of additional data.

Point clouds record the geometric characteristics of 3D objects, and points with geometric continuity in a local region belong to the same category. Thus, we can treat points with similar geometric characteristics as one unit and establish relationships between units, resulting in the conversion from unstructured 3D points to structured data. Through learning the context propagation in structured data from observations, we can predict the labels of units and adjust the graph structure due to the semantics. It results in a scene graph to describe the object-level relationships in the scene.

Specifically, we introduce more features to augment the geometric characteristics of 3D points. Thus, points with geometric continuity can be clustered into the same group, and the point cloud is converted into several point groups. Through modeling the relationships between groups, the initial structural representation is established.

A Dynamic Graph Generation Network (DGGN) is proposed to generate scene graphs according to the initial structural representation of point clouds. DGGN is a multi-level network and can adjust the graph structure dynamically. The group-level module predicts the semantic labels of groups with relationships by using Gated Recurrent Units (GRUs) to propagate contexts between groups. The point-level module further classifies points inner in ambiguous regions (the mixed group containing points of different categories). The DGGN splits and merges point groups according to the predicted semantics, finally generating a graph to describe the point-cloud-based scene.

The main contributions of this work can be summarized as follows:

• We first propose the framework to realize the graph generation of point clouds without using additional manually labeled graph structures. It represents the unstructured point cloud as structured data and generates scene graphs to describe the object-level relationships in the 3D scene.

• We propose DGGN to achieve the semantic prediction of targets with different granularity. It dynamically adjusts the graph structure according to the predicted semantics, resulting in a precise graph generation.

2 Related Work

2.1 Scene graph generation

The scene graph was first proposed in image-based tasks[5]. A graph is used to describe the semantic label of objects and their relationships in the image, and it helps applications such as image retrieval. Later, there are lines of works promoting the study of scene graph generations in 2D computer vision[4, 8, 9]. Different approaches are proposed to generate scene graphs according to the understanding of images, such as a variant of Graph Convolutional Network (GCN)[10] and Long Short-Term Memory (LSTM)-based MotifNet[11]. Most of these methods tackled the 2D graph generation problem with a pixel-level object detector extracting node/edge features. These detectors cannot be directly used for graph generation in a scene of 3D points.

Only a few works explored the graph generations in scenes of 3D points due to the lack of datasets. Reference [12] solved this problem and proposed 3RScan as the benchmark dataset for 3D scene graph representation learning[6, 7]. Reference [6] took use of PointNet and GCN to regress a graph from the scene of 3D points. After that, Ref. [7] put forward an improved Edge-oriented Graph Convolutional Network (EdgeGCN) to exploit multi-dimensional edge features for modeling object-level relationships in a 3D scene. All these works promote scene graph generation in point clouds. However, all these generation methods rely on the manually annotated class-agnostic graph structure given in the dataset. These methods do not work in the dataset without the graph structure. Thus, we explore solving this problem and achieving the scene graph generation from original 3D points without any other labeled graph structure.

2.2 Points-to-group mapping in structural representation

Points-to-group mapping is the basis of structural information extraction from unstructured 3D points. Relevant methods can be mainly divided into four categories: region-growing, edge-based, voxel-based, and clustering-based[13, 14].

The region-growing methods[15, 16] randomly select seeds for region growing. Regions with similar spatial features or surface properties are merged into the same group. This kind of method depends on seed selection excessively and demands a lot of computation. The edge-based methods[17, 18] map the points inside the same boundary obtained by edge detection as a group. These methods work well in simple scanned scenes. However, discontinuous boundaries often occur in large-scale scenes of 3D points. The voxel-based methods[19]

divide the point cloud into a few small voxels. This kind of work is of low efficiency because it needs so many voxels to describe a scene without losing geometrical characteristics. It requires very high computing resources for further graph generation. The clustering-based method[20, 21] effectively overcomes the other kinds of methods' shortcomings. It clusters points with similar features into the same group adaptively and automatically, which inspires us a lot.

3 Approach

A multi-level framework of point cloud graph generation is shown in Fig. 1. A clustering-based method converts the points into groups, resulting in an initial structural representation of the point cloud. Next, DGGN with multi-level semantic prediction and node splitting/merging functions outputs the final scene graph.

3.1 Initial structural representation

Structural representation of 3D point clouds is the basis of scene graph generation. To achieve this goal, we first enhance the geometric characteristics of every 3D point so that targets with similar features can be clustered together. Based on this, point groups with contextual relationships form the initial structural representation of the point cloud.

3.1.1 Feature augment

The original point cloud only provides discrete features, which cannot sufficiently describe the geometric characteristics of 3D points. To address this problem, we add the features illustrated in Table 1 to each 3D point.

In Table 1, F_1, F_2, and F_3 are widely used in the semantic or individual segmentation of outdoor scenes with obvious boundaries between objects[22], where λ_1, λ_2, and λ_3 ($\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$) are the eigenvalues of the covariance matrix formed by the coordinates of every point's k-nearest neighbors[17]. However, they do not work well in the segmentations of scenes with ambiguous boundaries. Reference [23] introduced F_4, F_5, and F_6 to enlarge the feature difference between objects of different categories, which helps to solve this problem. F_4 and F_5 represent the directionality of the local region around every point, where λ_{3x}, λ_{3y}, and λ_{3z}

Table 1 Geometric features.

Name	Formula	Description
F_1	$(\lambda_1 - \lambda_2)/\lambda_1$	Linearity
F_2	$(\lambda_2 - \lambda_3))/\lambda_1$	Planarity
F_3	λ_3/λ_1	Scattering
F_4	$\frac{1}{e^{-(\lambda_{3z}/\lambda_{3x})}+1}$	Directionality
F_5	$\frac{1}{e^{-(\lambda_{3z}/\lambda_{3y})}+1}$	Directionality
F_6	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$	Change of curvature

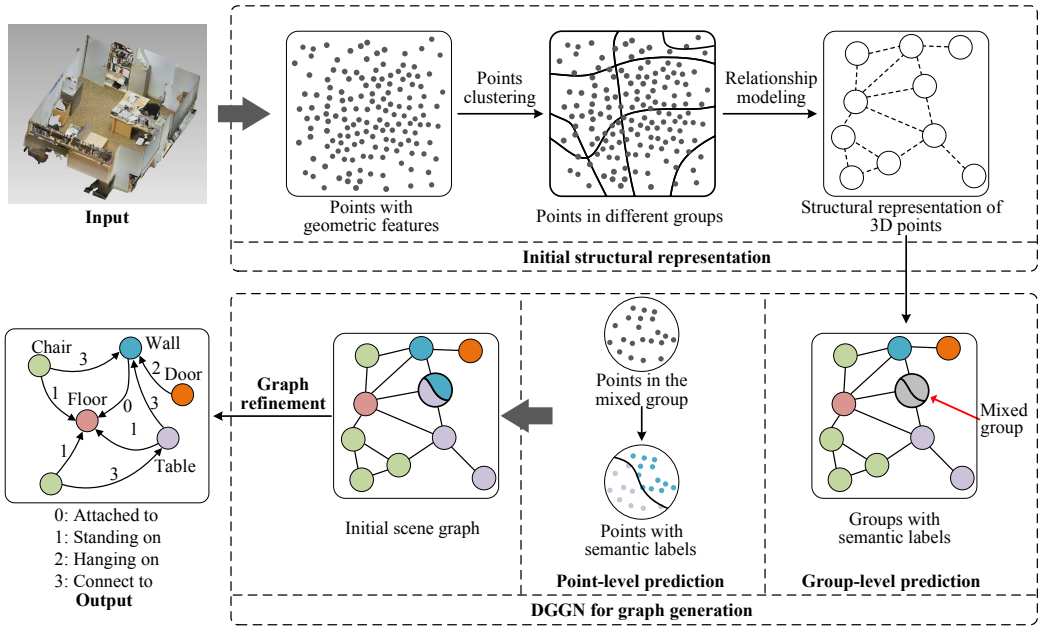


Fig. 1 Overall framework of our method consists of an initial structural representation module and a dynamic graph generation module. A point cloud is considered as the input, and the scene graph is the output. Points with similar geometric features are clustered into the same group, and the initial graph of the point cloud is achieved by establishing the relationships between groups. The group-level module predicts the group labels and judges whether the group is mixed (containing points of different categories or not). Every point in the mixed group is classified again with the point-level module. The final scene graph is generated through graph refinements.

are the x-component, y-component, and z-component of λ_3, respectively. F_6 represents the curving degree of the local region.

3.1.2 Points clustering and relationship modeling

This section discusses how to establish the initial structural representation of the point cloud. In other words, how to cluster points with similar geometric features into the same group and model the relationships between groups.

The original point cloud can be expressed as a feature metric $f \in \mathbf{R}^{V\times 6}$, where V is the total number of points, and 6 is the number of geometric features. $f_i \in \mathbf{R}^6$ indicates the feature vector of the i-th point. We aim to calculate an adjustable feature metric $h \in \mathbf{R}^{V\times 6}$ to describe the point groups, and $h_i \in \mathbf{R}^6$ indicates the feature vector of the i-th point. On the one hand, the distance between h and f should be minimized, thus retaining the point cloud's geometric features. It can be denoted as $\min \sum_{i\in V} \|h_i - f_i\|^2$, where $\|h_i - f_i\|^2$ indicates the distance between h_i and f_i. On the other hand, the feature vector of adjacent points in h should be as same as possible, which means the adjacent ones can be clustered together. Minimizing the Iverson bracket function $\delta(\cdot)$ helps to achieve this goal, where $\delta(h_i - h_j \neq 0) = 0$ when h_i equals to its adjacent point h_j. In summary, point clustering can be denoted as an optimization problem[24] as follows:

$$h^* = \arg\min_{h\in\mathbf{R}^{V\times 6}} \sum_{i\in V} \|h_i - f_i\|^2 + \rho \sum_{(i,j)\in e} \delta(h_i - h_j \neq 0) \tag{1}$$

where ρ is the weight coefficient used to balance the first and second parts of the equation. In solving this minimization problem, the first part promotes the solution h^* to approach f, and the second part promotes adjacent targets to have the same feature vector. ℓ_0-cut pursuit algorithm[25] helps to solve this problem.

In Eq. (1), $(i, j) \in e$ represents that point i is linked to point j with edge e. As the basis of the graph generation, every point should be able to route to any other point through the point-level relationships. If not, a point cloud will be split into several parts, resulting in several isolated graphs to represent a whole scene in the final generation. Delaunay triangulation[26] helps to establish these connecting relationships between points. It is widely used because of its efficient and stable data representation. This algorithm converts the whole point cloud into lots of edge-sharing triangles, in which the points correspond to the vertexes of triangles. On the one hand, triangles' edges represent the point-level relationship; on the other hand, all the points are within a whole. After points clustering, two groups are considered connected if the points in one group are linked to the other. Figure 2 illustrates the initial structural representation of a scene of the point cloud.

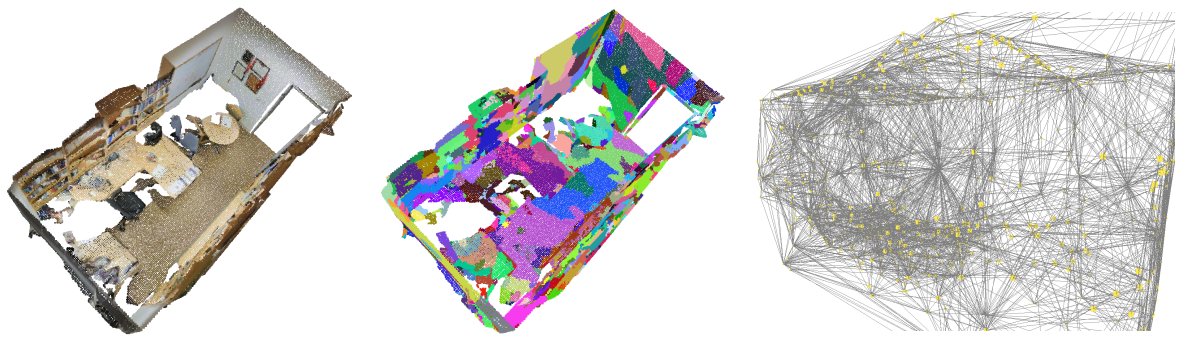
3.2 Dynamic graph generation

The initial structural representation achieves the group-level relationship establishment without semantics. Based on this, this section discusses establishing the semantical object-level relationship, namely the graph generation. We introduce multi-level modules to judge the semantic label of point groups; Then, groups with semantics are split and merged again, resulting in the final graph representation (see Fig. 3).

3.2.1 Multi-level semantic prediction

(1) Group-level semantic prediction

Each group contains a different number of points. It is necessary to have an expression of each group with a fixed dimension. To achieve this purpose, we introduce a simplified PointNet[27] to obtain a 32-dimensional



(a) (b) (c)

Fig. 2 Initial structural representation of a scene of the point cloud. (a) Original point cloud, (b) points after grouping (points in the same group with the same color), and (c) structural representation (a yellow dot indicates a point group, and a gray line indicates a connection relationship).

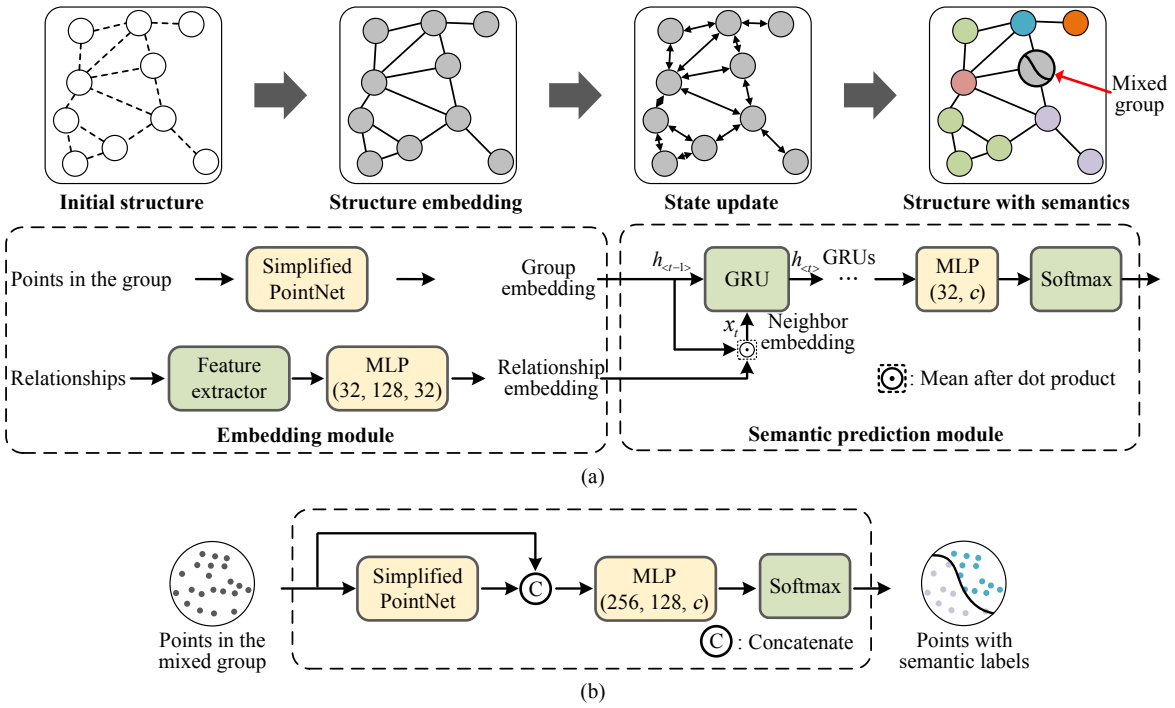


Fig. 3 Semantic prediction of different levels in DGGN. (a) Group-level semantic prediction. The structure embedding is calculated firstly. Then, GRU modules update the structure's state, considering the group embeddings and relationship embeddings. MultiLayer Perceptrons (MLPs) cooperate with Softmax to output the structures with semantics. (b) Point-level semantic prediction. For the points in the mixed group, the group embedding is concatenated to every point for the point-level semantic predictions. *c* denotes the category of class labels.

embedding describing every point group. We also introduce the shape-based, size-based, and point-based features proposed by Ref. [22] to describe the relationships between groups, cooperating with MLP to get a 32-dimensional relationship embedding.

For each point group, the semantic label is decided not only by itself but also by the effects of neighbors. We introduce GRU modules[28] to update the group state based on these factors, achieving the semantic prediction of each point group. Specifically, at each time step t, GRU updates the target's hidden state $h_{<t>}$, considering the hidden state $h_{<t-1>}$ at step $t-1$ and x_t as input, which can be denoted as

$$h_{<t>} = f(h_{<t-1>}, x_t),$$

where $f()$ indicates the GRU module. If we treat $h_{<t>}$ as the embedding of the target group itself and represent x_t as the neighbor effects, the GRUs can update the target's embedding step by step. Figure 3 illustrates the details; we use the mean value of the dot product of the group embedding and the relationship embedding to represent the neighbor effects. The group embeddings are updated iteratively, resulting in semantic prediction. Note that the group-level module also judges whether the group is mixed (containing points of more than two categories) or not, which supports the following point-level semantic prediction.

(2) Point-level semantic prediction

As discussed above, the point grouping module aims to cluster points with similar semantic labels. However, several groups inevitably contain points of different categories. It introduces errors to the final graph generation if treating the mixed group as a whole for semantic prediction. To address this problem, we introduce a simplified PointNet to predict the semantic label of each point again. The simplified PointNet retains the main body of the original PointNet[27], like the MLPs for point-wise feature learning and the Max-Pooling for global feature representation. Some auxiliary modules, such as the input/feature transformation, are removed. The point-level module only works on the points of the mixed group, accounting for a very small part of the whole point cloud. The performance improvement brought by the input/feature transformation modules is limited[27], influencing little to the final scene graph generation. The global feature of the point group obtained by the simplified PointNet is concatenated to every point, and MLPs work with Softmax to achieve point-level semantic label prediction.

(3) Loss function

Unlike the unsupervised initial structural representation, graph generation is a supervised process. Two cross-entropy loss functions guide the training of the group-level and the point-level module individually.

3.2.2 Graph refinement

A graph representation of point clouds with semantic labels has been established through multi-level semantic prediction. This section further discusses refining the graph, resulting in a plausible representation of given scenes. Adjacent point groups with the same label belong to the same object; points in the mixed group are of different categories. Thus, it is necessary to split and merge the point groups again. Besides, edge labels, as an important part of the graph representation, are needed to explain the relationships between objects.

As shown in Fig. 4, we first split the mixed group into different parts according to the semantic labels. Then, we merge groups with the same semantic label. Figure 5 illustrates the point group splitting and merging on a point cloud scene with our method. Based on this, we label the edges by querying the dictionary, which records the real-world relationship between adjacent objects, such as "a table standing on the floor". As a result, a reasonable final graph representation of a scene of the point cloud is established.

4 Experiment

4.1 Datasets

We introduce S3DIS, 3DSSG-O27R16, and Paris-Lille-3D to verify the performance of our methods, ranging from indoor to outdoor point cloud scenes.

(1) S3DIS[29] is a set of real-world indoor scenes of the point cloud, which contains 271 rooms belonging to 6 areas. This dataset includes 13 class labels and also records the points-to-object relationships. To achieve the supervised training and the testing evaluation for the

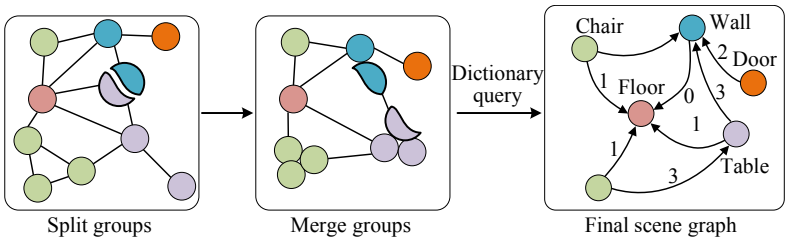


Fig. 4 Graph refinement. It splits mixed groups and merges groups with the same label. The final scene graph is obtained by introducing edge labels using the dictionary query.

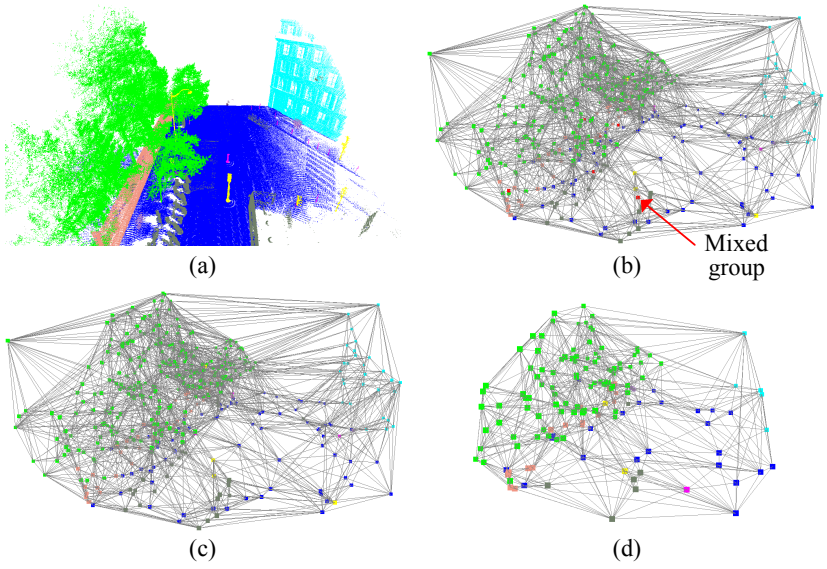


Fig. 5 Visualizations of point group splitting and merging. (a) An outdoor scene of the point cloud. Due to the lack of RGB information in the dataset, the points are colored according to the ground-truth labels for visual display. (b) The initial state of point groups with predicted labels. Dots with different colors denote point groups with different predicted labels, the red ones are for the mixed groups; The gray lines indicate the relationships between groups. (c) The mixed groups are split, then merged into the neighbor groups. (d) Groups with the same predicted label are further merged.

generation tasks, we make the scene graphs, in which nodes are labeled according to the semantic information of the corresponding objects, and the edges are labeled manually. Data in Area 5 are used for testing, while others for training.

(2) 3DSSG-O27R16[6, 7, 12] is widely used as the benchmark for graph generation tasks in point clouds. It contains 1318 scenes of 3D points and the corresponding scene graphs, including 27 object class labels and 16 relationship class labels. Our method aims at generating a scene graph, which records the precise relationships. Thus, we remove the ambiguous relationships in the dataset, such as "close by", retaining the precise contact relationships, such as "standing on". Random choosing 255 scenes for testing, while others for training.

(3) Paris-Lille-3D[30] is a large-scale urban outdoor point cloud dataset acquired by a mobile laser scanning system in France. It covers about 2 km of streets, containing more than 140 million points with 10 semantic classes. As same as the processing on S3DIS, the ground-truth scene graphs are labeled manually for the supervised training. Two scenes, Lille1 and Lille2, are used for training, while the scene Paris is for testing.

4.2 Implementation details

The weight coefficient ρ in Eq. (1) is set to 0.06. The training batch of DGGN is 2, and it is end-to-end trained by using back-propagation and Adam optimizer solver with a learning rate of 0.01. The point-level module works after ten epochs because it is easy to misclassify the mixed group early in training. The model is conducted by Pytorch on a single GeForce GTX 1080 Ti and an Intel (R) Core (TM) i7-7800X CPU. The point group of the training dataset is labeled as mixed if no more than 90% of the contained points belong to the same class. Otherwise, its label follows that of included points with the highest proportion.

4.3 Evaluation metrics

(1) Grouping Accuracy (GA) is designed to evaluate the accuracy of point grouping. It is defined in the following:

$$GA = \sum_{i=1}^{N} C_i / T \tag{2}$$

where C_i equals the number of points belonging to the class with the largest proportion. T and N are the numbers of points and point groups, respectively.

(2) Mean IoU (mIoU) denotes the mean intersection over a union of the containing 3D points of each node, which evaluates the performance of node generations.

(3) Top-k recall of relationship prediction (R@k-Rel) indicates the fraction of times of correct predictions in the top-k confident relationship predictions[31], which evaluates the performance of edge generations.

(4) Maximum Mean Discrepancy (MMD)[10, 32] indicates the distance between two data distributions. We introduce MMD based on degrees and clustering coefficients to quantify the similarity of the structures of generated graphs and those of the ground truths (the smaller, the better). It can be used to judge the rationality of the structure of the generated graphs.

4.4 Comparison with baselines

4.4.1 Baselines

Existing scene graph generation methods rely on given graph structure data, which are unfair to be treated as baseline methods. To address this problem, we embed the PointNet[27], KPConv[33], and GCN[34] into our framework as baselines. These widely used methods cooperate with our initial structural representation module to generate scene graphs for comparison.

Specifically, PointNet is a point MLP network, and KPConv is a point convolution one. They are all the neural networks for point-wise semantic prediction in point clouds. Thus, we use them to predict the semantic label of all the points in each point group, and each group's label follows that of included points with the highest proportion. Then, the final scene graphs are generated by splitting and merging groups, etc. Unlike PointNet and KPConv, GCN can directly work on the point clouds with structural representation. It replaces our method's multi-level semantic prediction module to generate scene graphs for comparison.

4.4.2 Quantitative results

(1) Results on S3DIS

Table 2 illustrates that our method outperforms the baseline methods in terms of all the metrics (compared with the 2nd ranking GCN: +2.0 mIoU, +3.7 R@20-Rel, – 0.05 MMD for degree, and + 0.02 MMD for cluster).

From the results, we can see that the point-based networks PointNet and KPConv perform worse than the graph-based networks (Ours and GCN). It is because

Table 2 Comparison results on S3DIS.

Embedded method	mIoU (%)	R@20-Rel(%)	MMD	
			Degree	Cluster
PointNet	49.7	57.1	0.72	0.32
KPConv	52.4	63.2	0.65	0.22
GCN	53.3	63.8	0.51	**0.17**
DGGN (Ours)	**55.3**	**67.5**	**0.46**	0.19

the point-based methods work independently on each point, which ignores the initial structural representation. The graph-based methods fully explored the contextual relationships using the structural information, thus leading to better graph generation results. Besides, our method performs the best due to the semantic modeling of different granularity. However, the GCN can only model the semantic information at the granularity of the point group, resulting in relatively poor performance.

(2) Results on 3DSSG-O27R16

Table 3 shows that our method performs the best (compared with the 2nd ranking GCN: + 1.5 mIoU, + 0.4 R@20-Rel, – 0.02 MMD for degree, and – 0.26 MMD for cluster) and also verifies the conclusions obtained from Table 2. Besides, there is a significant decline in all the metrics on 3DSSG-O27R16 than on S3DIS. 3DSSG-O27R16 contains more categories of objects and object-level relationships than S3DIS, posing much more difficulties in modeling structures and semantics of point-cloud-based scenes. Thus, it results in performance degradation of all the methods.

(3) Results on Paris-Lille-3D

Table 4 shows that our method performs best in terms of MMD. However, it does not work as well as the KPConv in terms of mIoU and R@20-Rel (compared with the KPConv: – 0.4 mIoU, – 0.5 R@20-Rel, – 0.02 MMD for degree, and – 0.14 MMD for cluster). The complexity of outdoor scenes poses challenges to graph generation, thus leading to obvious performance degradation for lots of methods.

The structural representation becomes complicated due to the scenes' complexity, as shown in Fig. 5. Besides, the variety of outdoor scenes cause the diversities of representations. All these characteristics bring difficulties for the methods based on the structural representation of point clouds. It makes our method misjudge the labels of lots of groups, leading to a decline in mIoU and R@20-Rel. However, our graph-based network focuses on learning the contextual relationships between groups, retaining a good semantic continuity between point groups. It cooperates with the group splitting and merging function to obtain good graph structures, thus, performing well in terms of MMD.

4.4.3 Qualitative results

Figure 6 shows the graph generations of several

Table 3 Comparison results on 3DSSG-O27R16.

Embedded method	mIoU (%)	R@20-Rel (%)	MMD	
			Degree	Cluster
PointNet	38.6	42.9	0.83	0.65
KPConv	41.5	41.7	0.98	0.54
GCN	42.1	47.8	0.80	0.72
DGGN (Ours)	**43.6**	**48.2**	**0.78**	**0.46**

Table 4 Comparison results on Paris-Lille-3D.

Embedded method	mIoU (%)	R@20-Rel (%)	MMD	
			Degree	Cluster
PointNet	35.4	37.5	0.98	0.72
KPConv	**47.1**	**52.2**	0.73	0.66
GCN	41.3	49.3	0.93	0.75
DGGN (Ours)	46.7	51.7	**0.71**	**0.52**

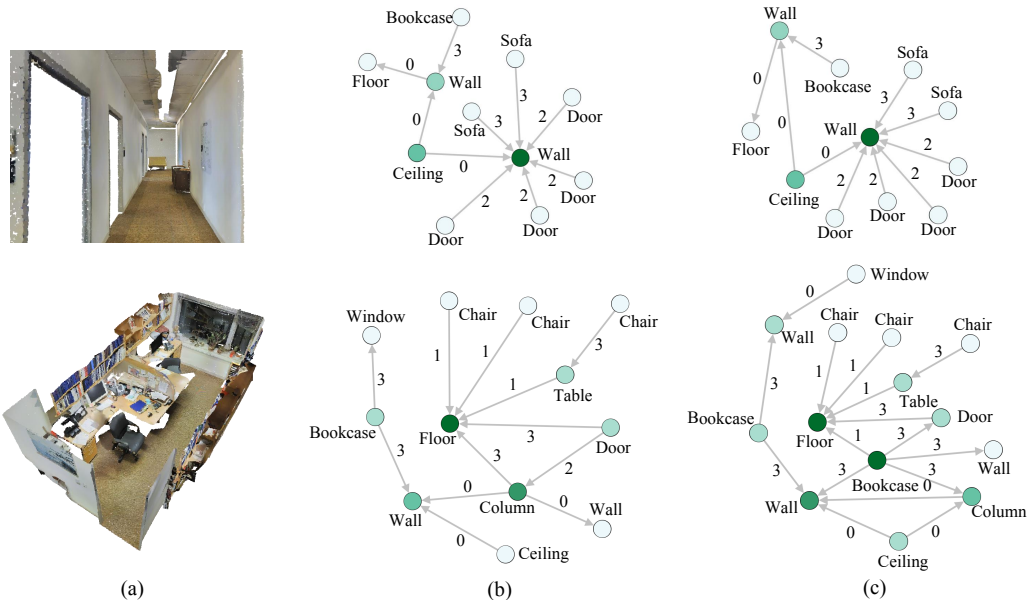


Fig. 6 Graph generations from several scenes in S3DIS (0: attached to; 1: standing on; 2: hanging on; 3: connect to). (a) Point-cloud-based 3D scene, (b) graph generations using our method, and (c) ground truth.

randomly selected scenes in S3DIS Area 5. These graphs describe the objects with relationships in the point cloud scene, such as "a bookcase connected to the wall". Our generation is similar to the ground truth of the scene in the first row. However, mistakes happen in the more complex scene in the second row. For example, our generated graph contains only one bookcase, but there are two in the point cloud scene. The reasons may be as follows: the initial structural representation splits the point cloud into several groups, and groups with bookcase points are similar to those with wall points. It makes our DGGN misjudge the semantic information of these groups, resulting in only one bookcase in the generated graph.

Figure 7 illustrates the graph generation from a split scene of Paris-Lille-3D. The graphs are more complex than those generated from indoor scenes. The structure of the generated graph is similar to the ground truth. However, the labels of some objects are misjudged or missing, such as some pedestrians are missing in the scene graph. As we explain above, our method misjudges some point groups' labels, leading to mistakes in the final scene graph.

To explore the effectiveness of our method in unfamiliar items, we conduct new experiments on scenes of the point cloud collected by our 3D laser scanner. Figure 8 gives the scene graphs generated by our method, denoting the objects that existed in the office and the

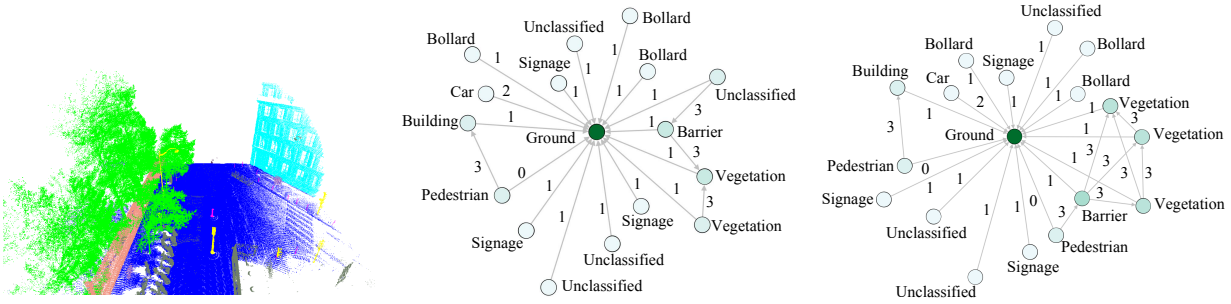


Fig. 7 Graph generations from a split scene of Paris-Lille-3D. Some simple relationships between objects, such as "bollard standing on the ground", are widespread in the scene. We only reserve a few representative ones to make the graph more intuitive (0: walking on; 1: standing on; 2: driving on; 3: connect to). (a) Point-cloud-based 3D scene, (b) graph generations using our method, and (c) ground truth.

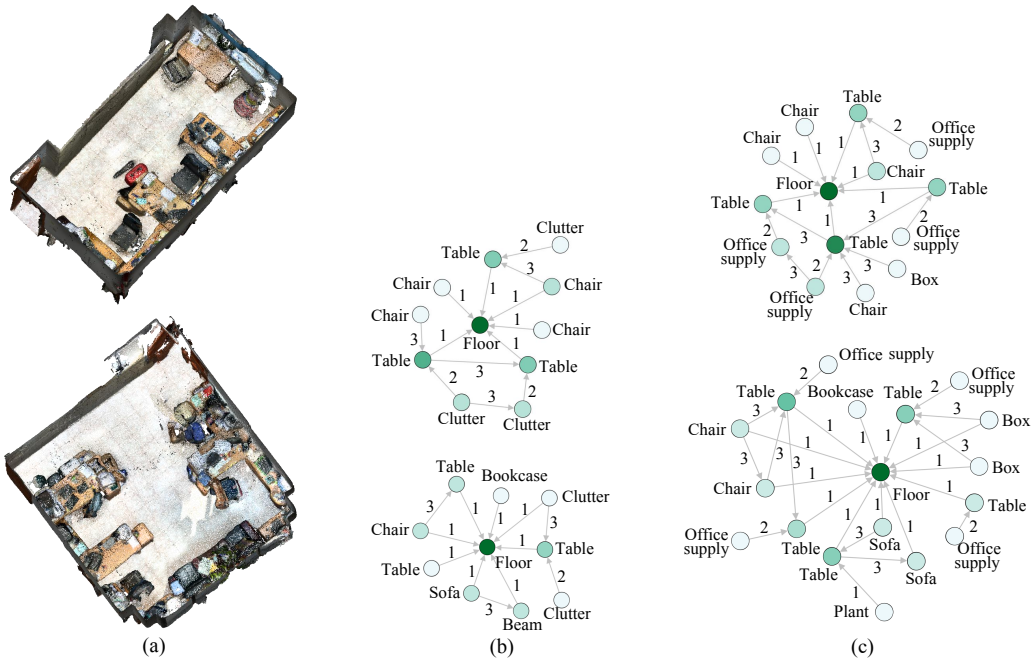


Fig. 8 Graph generations from unfamiliar scenes. "Walls", "doors", and "windows", are removed for brief visualizations (1: standing on; 2: lying on; 3: connect to). (a) Point-cloud-based 3D scene, (b) graph generations using our method, and (c) ground truth.

relationships between them. However, they are not as precise as the ground truth scene graphs. For example, the object-level relationship "office supply lying on the table" is misjudged as "cluster lying on the table". The difference in object semantics between training and test scenes causes this problem. Specifically, in the training phase, our method has never observed the objects labeled as "office supplies". Thus, our method misjudges many objects in the testing phase, leading to many ambiguous object-level relationships in the final generated graphs.

4.5 Ablation study

We remove or change several designs individually, exploring their effect on the initial structural representation and the final graph generation.

4.5.1 Effect of geometric features

Table 5 illustrates that using all the features achieves the best point grouping performance. Randomly dropping the features lead to a decline in the accuracy of point grouping. It indicates that these geometric features help cluster points with similar semantic labels better, which will benefit the performance of final graph generation.

4.5.2 Effect of the weight coefficient in Eq. (1)

Table 6 denotes that setting ρ to 0.06 achieves a good balance between the grouping accuracy and the mean group numbers. Setting ρ to 0.04 obtains the best accuracy due to a weakened strength to cluster more points into a group. However, it increases the number of point groups and will result in high computational resources for graph generations. Setting ρ to 0.08 shows the opposite situation. Thus, 0.06 is a more appropriate approach scheme for point grouping in the initial structural representation learning.

Table 5 Ablation study of the geometric features on point grouping for S3DIS.

Features selection				GA (%)
F_1, F_2, F_3	F_4	F_5	F_6	
✓	–	–	–	76.5
✓	✓	–	–	80.1
✓	–	✓	–	77.5
✓	–	–	✓	83.2
✓	–	✓	✓	83.7
✓	✓	–	✓	90.3
✓	✓	✓	–	88.4
✓	✓	✓	✓	**91.9**

Table 6 Ablation study of the weight coefficient in Eq. (1) on point grouping for S3DIS.

ρ	GA (%)	Mean group number
0.04	93.5	3824
0.06	**91.9**	**1211**
0.08	89.6	417

4.5.3 Effect of the group-level module in DGGN

The group-level module updates the states of point groups for semantic prediction by propagating information among neighbors. To approve its effectiveness, we remove this module and only use the point-level module to judge the semantic label of nodes. Table 7 shows that removing the group-level module achieves poor graph generation performance (compared with the original network: –11.7 mIoU, –14.6 R@20-Rel, + 0.35 MMD for degree, and + 0.41 MMD for cluster). It indicates that modeling the contextual relationships between point groups contribute a lot to the graph generation, verifying the effect of the group-level module.

4.5.4 Effect of the point-level module in DGGN

The point-level module helps to reduce the errors introduced by the initial structural representation learning. Table 7 illustrates that removing the point-level module also leads to a decline in the graph generation performance (compared with the original network: − 1.2 mIoU, − 0.3 R@20-Rel, + 0.03 MMD for degree, + 0.09 MMD for cluster). The impact of the point-level module on graph generation is not as large as that of the point-level one. It only works on limited point groups, resulting in a limited performance increase. However, it helps to improve the precision of generated graphs.

5 Conclusion

This paper presents a method to generate the graph of scenes of the point cloud. Unlike other works of scene graph generation on the point clouds, which rely on the manually labeled graph structure data. Our method automatically outputs scene graphs without additional data. It models the geometric features and clusters

Table 7 Ablation study of the group-level module and the point-level module in DGGN. RGM stands for removing the group-level module, and RPM stands for removing the point-level module.

Embedded method	mIoU (%)	R@20-Rel (%)	MMD	
			Degree	Cluster
RGM	43.6	52.9	0.81	0.60
RPM	54.1	67.2	0.49	0.28
Original	**55.3**	**67.5**	**0.46**	**0.19**

3D points into groups, resulting in an initial structure representation of the point cloud. Besides, a DGGN network containing different levels of modules achieves the final graph construction of point-cloud-cased scenes. Experiments prove that our method outperforms other baseline methods, outputting reliable graphs to describe 3D scenes. Our method can be used to promote point-cloud-based applications, such as object retrieval in a 3D space.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62173045 and 61673192), the Fundamental Research Funds for the Central Universities (No. 2020XD-A04-2), and the BUPT Excellent PhD Students Foundation (No. CX2021222).

References

[1] A. A. Liu, H. Zhou, W. Nie, Z. Liu, W. Liu, H. Xie, Z. Mao, X. Li, and D. Song, Hierarchical multi-view context modelling for 3D object classification and retrieval, *Inf. Sci.*, vol. 547, pp. 984–995, 2021.

[2] L. Deng, M. Yang, Z. Liang, Y. He, and C. Wang, Fusing geometrical and visual information via superpoints for the semantic segmentation of 3D road scenes, *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 498–507, 2020.

[3] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, Graph R-CNN for scene graph generation, in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 690–706.

[4] X. Li and S. Jiang, Know more say less: Image captioning based on scene graphs, *IEEE Trans. Multimedia*, vol. 21, no. 8, pp. 2117–2130, 2019.

[5] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, Scene graph generation by iterative message passing, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 3097–3106.

[6] J. Wald, H. Dhamo, N. Navab, and F. Tombari, Learning 3D semantic scene graphs from 3D indoor reconstructions, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 3960–3969.

[7] C. Zhang, J. Yu, Y. Song, and W. Cai, Exploiting edge-oriented reasoning for 3D point-based scene graph analysis, in *Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2021, pp. 9700–9710.

[8] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling, Scene graph generation with external knowledge and image reconstruction, in *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 1969–1978.

[9] Z. Lin, F. Zhu, Q. Wang, Y. Kong, J. Wang, L. Huang, and Y. Hao, RSSGG_CS: Remote sensing image scene graph generation by fusing contextual information and statistical knowledge, *Remote Sens.*, vol. 14, no. 13, p. 3118, 2022.

[10] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec, GraphRNN: Generating realistic graphs with deep auto-regressive models, in *Proc. 35th Int. Conf. Machine Learning*, Stockholm, Sweden, 2018, pp. 5694–5703.

[11] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, Neural motifs: Scene graph parsing with global context, in *Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 5831–5840.

[12] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Nießner, RIO: 3D object instance Re-localization in changing indoor environments, in *Proc. 2019 IEEE/CVF Int. Conf. Computer Vision*, Seoul, Republic of Korea, 2019, pp. 7657–7666.

[13] Y. Xie, J. Tian, and X. X. Zhu, Linking points with labels in 3D: A review of point cloud semantic segmentation, *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 4, pp. 38–59, 2020.

[14] W. Liu, Z. Liu, Q. Li, Z. Han, and A. Núñez, High-precision detection method for structure parameters of catenary cantilever devices using 3-D point cloud data, *IEEE Trans. Instrum. Meas.*, vol. 70, p. 3507811, 2021.

[15] J. Xiao, J. Zhang, B. Adler, H. Zhang, and J. Zhang, Three-dimensional point cloud plane segmentation in both structured and unstructured environments, *Rob. Auton. Syst.*, vol. 61, no. 12, pp. 1641–1652, 2013.

[16] J. E. Deschaud and F. Goulette, A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing, presented at 3D Data Processing Visualization and Transmission, Paris, France, 2010.

[17] T. Rabbani, F. Van Den Heuvel, and G. Vosselman, Segmentation of point clouds using smoothness constraints, in *Proc. ISPRS Commission V Symp.: Image Engineering and Vision Metrology*, Dresden, Germany, 2006, pp. 248–253.

[18] M. A. Wani and H. R. Arabnia, Parallel edge-region-based segmentation algorithm targeted at reconfigurable MultiRing network, *J. Supercomput.*, vol. 25, no. 1, pp. 43–62, 2003.

[19] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, Voxel cloud connectivity segmentation-supervoxels for point clouds, in *Proc. 2013 IEEE Conf. Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013, pp. 2027–2034.

[20] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, Fast approximate energy minimization with label costs, *Int. J. Comput. Vis.*, vol. 96, no. 1, pp. 1–27, 2012.

[21] D. Kong, L. Xu, X. Li, and S. Li, K-plane-based classification of airborne LiDAR data for accurate building roof measurement, *IEEE Trans. Instrum. Meas.*, vol. 63, no. 5, pp. 1200–1214, 2014.

[22] L. Landrieu and M. Simonovsky, Large-scale point cloud semantic segmentation with superpoint graphs, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 4558–4567.

[23] C. Qi and J. Yin, Multigranularity semantic labeling of point clouds for the measurement of the rail tanker component

with structure modeling, *IEEE Trans. Instrum. Meas.*, vol. 70, p. 5000312, 2021.
[24] S. Guinard and L. Landrieu, Weakly supervised segmentation-aided classification of urban scenes from 3D LiDAR point clouds, in *Proc. Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Hannover, Germany, 2017, pp. 151–157.
[25] L. Landrieu and G. Obozinski, Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs, *SIAM J. Imaging Sci.*, vol. 10, no. 4, pp. 1724–1766, 2017.
[26] L. P. Chew, Constrained Delaunay triangulations, *Algorithmica*, vol. 4, nos. 1–4, pp. 97–108, 1989.
[27] R. Q. Charles, S. Hao, K. Mo, and L. J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 77–85.
[28] K. Cho, B. van Merrienboer, A. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014, pp. 1724–1734.
[29] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, 3D semantic parsing of large-scale indoor spaces, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 1534–1543.
[30] X. Roynard, J. E. Deschaud, and F. Goulette, Paris-lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification, *Int. J. Rob. Res.*, vol. 37, no. 6, pp. 545–557, 2018.
[31] C. Lu, R. Krishna, M. S. Bernstein, and L. Fei-Fei, Visual relationship detection with language priors, in *Proc. 14^{th} European Conf. Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 852–869.
[32] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, A kernel two-sample test, *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, 2012.
[33] H. Thomas, C. R. Qi, J. E. Deschaud, B. Marcotegui, F. Goulette, and L.J. Guibas, KPConv: Flexible and deformable convolution for point clouds, in *Proc. 2019 IEEE/CVF Int. Conf. Computer Vision*, Seoul, Republic of Korea, 2019, pp. 6410–6419.
[34] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, presented at the 5th Int. Conf. Learning Representations, Toulon, France, 2017.

Chao Qi received the MEng degree from Beijing Jiaotong University, China in 2014. He is currently a PhD candidate at the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. He is also an assistant research fellow at the Standard and Metrology Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing, China. His research interests include deep learning, pattern recognition, and 3D vision.

Jianqin Yin received the PhD degree from Shandong University, China in 2013. She is currently a professor at the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include service robot, pattern recognition, machine learning, and image processing.

Zhicheng Zhang received the BEng degree in information engineering, the MEng degree in pattern recognition, and the PhD degree in control theory and engineering from Jilin University, China in 2005, 2007, and 2011, respectively. He is currently an associate professor at the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include signal processing, artificial intelligence, and computational intelligence.

Jin Tang received the PhD degree from Beijing Institute of Technology, China in 2007. She is currently an assistant professor at the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China. Her research interests include signal processing, pattern recognition, and deep learning.