The Visual Computer (2020) 36:2407–2418
https://doi.org/10.1007/s00371-020-01892-8

ORIGINAL ARTICLE

PGCNet: patch graph convolutional network for point cloud segmentation of indoor scenes

Yuliang Sun[1] · Yongwei Miao[2] · Jiazhou Chen[1] · Renato Pajarola[3]

Published online: 14 July 2020

Abstract
Semantic segmentation of 3D point clouds is a crucial task in scene understanding and is also fundamental to indoor scene applications such as indoor navigation, mobile robotics, augmented reality. Recently, deep learning frameworks have been successfully adopted to point clouds but are limited by the size of data. While most existing works focus on individual sampling points, we use surface patches as a more efficient representation and propose a novel indoor scene segmentation framework called patch graph convolution network (PGCNet). This framework treats patches as input graph nodes and subsequently aggregates neighboring node features by dynamic graph U-Net (DGU) module, which consists of dynamic edge convolution operation inside U-shaped encoder–decoder architecture. The DGU module dynamically update graph structures at each level to encode hierarchical edge features. Incorporating PGCNet, we can segment the input scene into two types, i.e., room layout and indoor objects, which is afterward utilized to carry out final rich semantic labeling of various indoor scenes. With considerable speedup training, the proposed framework achieves effective performance equivalent to state-of-the-art for segmenting standard indoor scene dataset.

Keywords Point cloud · Scene segmentation · Surface patch · Graph convolutional network · Edge convolution · Encoder–decoder

1 Introduction

3D indoor scene understanding requires a thorough analysis on geometric and semantic context of interior scene. Indoor scene semantic segmentation, in which indoor objects are assigned with different labels, is a fundamental sub-task of scene understanding. Point cloud, which can be acquired directly by most depth scanning devices, is a common geometric representation in the literature of computer graphics and computer vision [1–4]. Point cloud segmentation of indoor scenes is now attracting growing attention because of its various applications such as virtual/augmented reality [5], mobile robotics [6], indoor navigation [7].

✉ Yongwei Miao
ywmiao2009@hotmail.com

1 College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China

2 College of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China

3 Department of Informatics, University of Zurich, 8050 Zurich, Switzerland

Semantic segmentation of indoor scenes is still challenge due to incomplete raw inputs, the scale of point cloud data, cluttered and always heavily occluded settings such as real-world indoor environments. For effectively processing point clouds, the essential issue is how to effectively extract the feature information of point cloud scenes or 3D shapes. Conventionally, handcrafted features of point clouds are chosen to analyze 3D geometry but they are difficult to select for specific tasks [8]. Now, deep learning has achieved significant success on 2D image processing, inspiring works in 3D space. Most previous methods address 3D geometry and 3D vision problems using voxels [9] or multi-view images [10] as input of convolutional neural networks (CNNs). The conversion from discrete point clouds to such regular representations is always time consuming. Recently, CNN architecture has been applied directly on point clouds to capture geometric features [1], using max pooling layer that acts as a symmetric function to encode global features from a group of points. Since this pioneering work does not fully consider local features, PointNet++ [11] further explores local information at multiple scales but still does not consider relationships between sampling points. In a parallel

development, Graph convolutional networks (GCNs) extend CNN architecture to graphs with irregular data. Since point cloud is a kind of unstructured data, GCNs have promising potential to consume unordered 3D discrete point clouds [12,13]. These methods treat each sampling point as a node in graph and show promise for implementing convolution on each node and its neighbors either in spectral or spatial domain. Most deep learning based methods focus on the extraction of intrinsic features for individual points and their neighbors within bounded range. Moreover, when applied to indoor scene understanding, these methods have to partition the whole point cloud scenes into blocks with limited point size.

To tackle the large-scale point cloud data and computationally time-consuming issues in deep learning based indoor point cloud segmentation, our presented method adopts surface patches as data representation. A fundamental observation in various cluttered indoor environments is that manmade objects are always constructed in a highly structured style, with a combination of various surface patches. Apart from dominant room layouts, the surface patches can also be identified in indoor furniture such as tables, chairs, and cabinets. The discrete sampling points in the surface patches remain geometrically consistent and can be considered as object parts. Taking surface patches as data representation for deep learning based segmentation task can largely alleviate the difficulties due to large scales of input data in a typical indoor scene with millions of sampling points, thus speed up network training. Besides, the surface patches of indoor objects generally have prior contextual information that can be easily captured. While the surface patches data cannot be directly used as the input of conventional CNNs, they can be treated as nodes in a graph structure and their geometric properties can be used as node features. The contextual relationships between pairwise patches can be regarded as edge features. To better extract local features and aggregates neighboring information, a Scene Patch Graph (SPG) is constructed. Considering surface patches as nodes and their spatial relationships as edges in SPG, our intention is to incorporate the patch graph convolutional network (also called PGCNet) framework for semantic segmentation of indoor point cloud scenes. To achieve this goal, we utilize a novel module dubbed dynamic graph U-Net (DGU) which incorporates dynamic edge convolution operation inside a U-shape encoder–decoder. Our DGU consists of multi-layer encoding blocks and corresponding decoding blocks with skip-connections. At each level of DGU, graph structure is updated through a Dynamic Edge Layer (DEL) which calculates the edges using pooled atrous k-nearest neighbor (k-NN), and edge features can be generated by an Edge Convolutional Layer (ECL). Given an indoor scene point cloud, surface patches are first generated by region growing based on similar normal and short

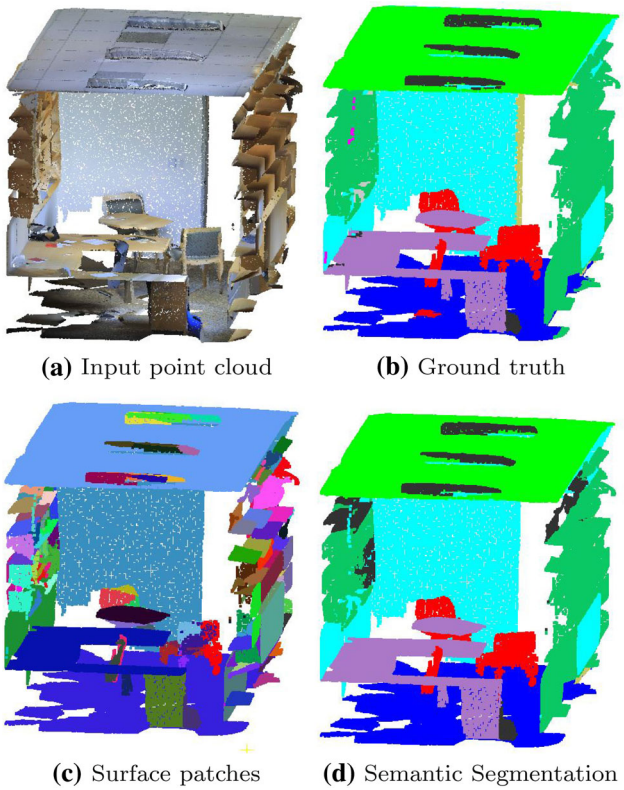


Fig. 1 Given an indoor scene point cloud (**a**), surface patches (**c**) are used as data representation. Based on our PGCNet, semantic segmentation (**d**) is performed

distance, see Fig. 1c. The SPG can also be constructed by extracted surface patches and their spatial relationships, such as adjacency. DGU is then employed to encode hierarchical edge features of the created SPG. Finally, A multi-layer perceptron (MLP) classifier is utilized to produce labels for each node in our SPG. Comparing with the existing methods that not distinguish the indoor objects from room layout, our novel framework introduced in this paper can be utilized to segment the input indoor scenes into two basic components, i.e., room layout and indoor objects, which can be further incorporated to generate rich semantic segmentation of indoor point cloud scenes, see Fig. 1d.

Contributions The main contributions of our work are summarized as follows,

- A novel framework, named as PGCNet, is introduced which takes the surface patches as data representation of indoor scene point clouds. This representation can largely reduce the data size and also enable us to apply graph convolutional network architecture on such large-scale point clouds.
- A new module DGU is utilized to capture and aggregate hierarchical edge features. This module employs

dynamic edge construction and edge convolution at each encoding block.
– With the help of our PGCNet, a layout-object classification network is trained first and can be further trained for semantic segmentation of indoor point cloud scenes. The network training is effective and the performance is competitive on segmenting standard indoor scenes Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [14] and ScanNet [15].

2 Related work

Indoor point cloud segmentation Previous solutions to indoor scene segmentation mainly include model-based and primitive-based schemes. With the help of pre-trained object categories, Nan et al. [16] described a search-classify pipeline for scene modeling. Li et al. [17] proposed an object-retrieved approach to replace scanned data with objects from 3D shape databases. Shi et al. [18] trained a set of classifiers for both objects and objects groups to decompose indoor subscenes. Although exploiting the information from databases, these methods highly relies on the diversity and the size of current 3D models datasets. Another strategy is to employ primitive-based approaches that will decompose indoor scenes into a set of geometric primitives. Mattausch et al. [19] presented an approach to segment indoor scenes by detecting repeated objects acquired from multi-rooms indoor scanning data. They adopted a set of nearly planar patches for representing indoor scenes, which can be clustered using patch similarity matrix based on the extracted shape geometrical descriptors. Recently, Hu et al. [20] proposed an 3D semantic segmentation approach using patch clusters as data representation. Inspired by these works, our method adopts surface patches as intermediate representation and utilizes them to construct a graph data.

3D deep learning based segmentation 3D deep learning is a hot research topic and has been shown great potential in semantic segmentation tasks [21,22]. Due to the success of CNNs on regular domain, previous work always adopted the view-based [10] or volumetric representations [21,23] to transfer 3D point clouds into grid structure data. For 3D object detection, Zhou et al. [23] transformed a set of points within each voxel into a feature vector. Tchapmi et al. [21] applied a fully CNNs to produce coarse voxel labels for semantic scene segmentation. By fusing multimodal inputs together, Hou et al. [24] jointly learned both image features with 3D geometry features for 3D instance segmentation. However, the data representation of multi-view images will be limited to scalability due to occlusion and large network input, and the voxel representation is also restricted due to its requirement for extra high dimension. Qi et al. [1] introduced the pioneer work which has a significant impact on employing CNNs directly on point cloud data. They further improved the performance by exploring local information [11]. Recently, there has been a surge of interest in leveraging GCNs to point cloud data. Point clouds and their neighboring information can be represented as a type of graphs which can utilize graph convolution for extracting local information. To segment 3D point cloud data, Wang et al. [12] proposed a dynamic edge convolution for updating the node adjacency at each feature map. Landrieu and Simonovsky [25] adopted superpoint as data representation and employed gated graph convolution network for point cloud segmentation. To further improve superpoint generation, Landrieu and Mohamed [26] proposed a supervised learning network for point cloud oversegmentation. Instead of recurrent-based GCNs in [25] that uses same graph to update hidden information, our method applies dynamic edge convolution at multi-level encoder to exploit local neighboring features.

GCNs GCNs are popular in current research as they can process convolution operations effectively for irregular discrete data [27]. Existing GCNs can be classified into spectral methods and spatial approaches. Spectral graph convolutional methods [28,29] use graph Laplacian eigenvectors and can conduct convolution operation on spectral features. Different from spectral-based schemes, spatial graph convolution methods utilize spatial information and aggregator to generate neighborhood feature embedding. The aggregator that passes node features message between neighbors can be LSTM based [30], attention based [31], or max-pooling based [12].

3 PGCNet-based point cloud segmentation of indoor scenes

The main object of this paper is to perform indoor scene semantic segmentation via surface patch representation. To this end, a novel framework known as PGCNet is introduced. The key components of PGCNet based indoor scene semantic segmentation are shown in Fig. 2. Given indoor scene point clouds, we first extract surface patches from input point clouds by region growing strategy and compute feature descriptors for each surface patch. Then an initial SPG can be constructed using the extracted surface patches and their pairwise relationships. DGU module consumes the initial SPG and encodes multi-layer edge features through dynamic graph construction and edge convolution. Take advantage of our DGU module, PGCNet finally outputs semantic labels for each surface patch of indoor scenes.

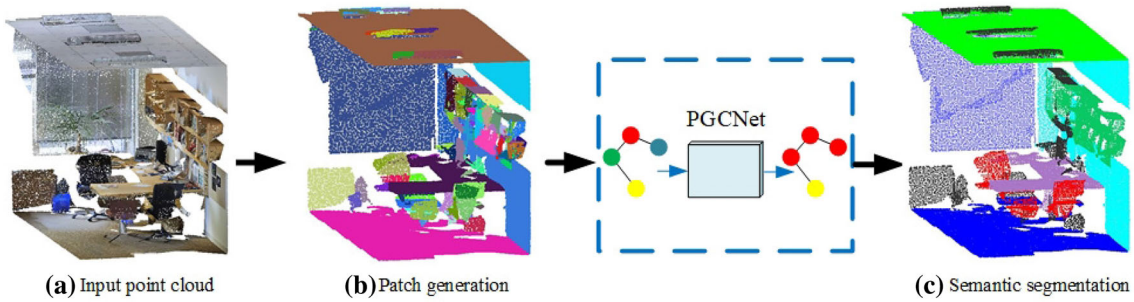


Fig. 2 The pipeline of our point cloud segmentation method. Given input point cloud (**a**), surface patches are generated (**b**) and fed into PGCNet. PGCNet finally produces semantic labels (**c**)

3.1 Scene patch graph (SPG) generation

Recent work [11,12] employed deep learning framework on point cloud data. Most of them applied the neural networks directly on discrete sampling points which may face expensive computation problem if the size of indoor scene point clouds is large. Partitioning whole indoor scene into blocks with limited number of points is one strategy to deal with such large-scale input data. The drawback of this solution is that the size of blocks and the number of sampling points are not easy to determine. The unsuitable selection of block size may limit the effectiveness of learning context information. Using surface patch representation can tackle these problems and thus provide another plausible solution to indoor scene segmentation.

Patch representation Surface patch structures are widely applied in 3D indoor scenes applications such as SLAM reconstruction [32], scene completion [33], room detection [34]. Manmade objects in indoor scenes are commonly constructed in a highly structured style, with a combination of planes. Apart from dominant room layouts, the planar surfaces can also be identified in indoor furniture such as tables, chairs and cabinets. Using surface patch as data representation of indoor scenes has several advantages. First, it can alleviate computation cost since the number of surface patches is generally smaller than sliding blocks. Second, the points in the same surface patch are geometrically consistent and can be considered as a whole. In addition, surface patch-based partition of whole scene is more reasonable than using uniformed sliding blocks, since it takes object shape structure into account. Furthermore, it is beneficial to learn from the relationships between neighboring surface patches that can be easily captured.

Patch growing Given an indoor scene point cloud as input, the goal of this step is to convert the input point cloud into surface patches. An efficient region growing strategy followed by [19] is applied to partition the input point cloud. Specifically, region growing for patch $\mathbf{P_i}$ starts from a seed point $\mathbf{s}$ selected from unassigned curvature ascending points list. Given a new closest neighbored point $\mathbf{p}$ outside $\mathbf{P_i}$, $\mathbf{p}$ can be added to $\mathbf{P_i}$ if the following conditions are satisfied:

$$\begin{aligned} \mathbf{n_p} \cdot \mathbf{n_s} &> t_1 \\ (\mathbf{p} - \mathbf{s}) \cdot \mathbf{n_s} &< t_2 \\ (\mathbf{p} - \mathbf{q}) \cdot \mathbf{n_q} &< t_3 \end{aligned} \tag{1}$$

here $\mathbf{q}$ is the last added point inside patch $\mathbf{P_i}$; $\mathbf{n_p}$, $\mathbf{n_s}$ and $\mathbf{n_q}$ are the normal of point $\mathbf{p}$, $\mathbf{q}$ and $\mathbf{s}$, respectively. These conditions specify the constraint that point $\mathbf{p}$ should be close to the patch surface with similar normal of point $\mathbf{s}$. The near planar patches, which represent the major structures of most indoor objects, are clustered preferentially according to seed selection and growing criteria. Some parts of organic-shaped objects can also be discovered at the end of patch generation and they are remained as nodes of SPG.

Patch descriptors The input point cloud is now represented by a set of surface patches. Feature descriptors are calculated to characterize each extracted patch. For each surface patch, the fitting rectangle is generated through projecting the bounding box onto dominant axes. The dominant axes could be determined by PCA method. For each fitting rectangle, the features descriptors includes centroid point position, PCA normal, height, length, area, color, ratio of length to width, fill ratio of convex hull area to area; see Table 1. In addition, boundary points of each patch are also measured.

Scene patch graph construction The initial SPG denoted $G^{(l_0)}(V, E)$ is constructed, where V is the set of surface patches and E represents the pair-wise patch spatial relations. Given a set of surface patches generated from input poindt cloud $\mathbf{P} = \{\mathbf{P_1}, \mathbf{P_2}, \ldots, \mathbf{P_N}\} \subseteq R^F$, where N is the number of surface patches and F is the number of features for each patch. Specifically, patch descriptors mentioned above are employed and combined as node features, which indicate compressed representation of input patches. In 3D space,

Table 1 Feature descriptors of a fitting rectangle

Feature	Description
P_p	Centroid position
P_n	PCA normal
P_h	Height
P_r	Ratio of length to width
P_a	Area
P_f	Fill ratio of convex hull area to P_a
P_c	Color

adjacent patches are more likely to be in the same object. To aggregate information from spatial surroundings, SPG is initialized by adding edges between two adjacent surface patches. Here, two surface patches are considered as adjacent based on the minimum distance of boundary.

3.2 Dynamic edge convolution

Here, our presented PGCNet adopts dynamic graph convolution since it can capture semantically similar structures at each feature embedding, even they may be distant in the original input graph. Our dynamic graph convolution consists of two types of layer, that is, DEL and ECL: DEL recalculates the edges using atrous K-NN followed by a TopK pooling; ECL generates edge features that depict the relationships between a node and its neighbors.

Dynamic graph construction Graph is a non-structural data which models a set of nodes and their relationships. GCNs generalize CNNs framework to graphs and consume irregular data. Spectral-based GCNs usually require eigen-decomposition of Laplacian matrices that defines the graphs. The computation cost of this kind of graph construction is generally expensive. Spatial-based GCNs implement convolution operation on node spatial relation-based graphs. This type of graph construction has more flexible form than fixed graph Laplacian matrices. The pioneering work [1] employs CNNs directly on point cloud. Its succeeding work [11] further considers local structures using farthest point sampling to select nodes from fixed input points. While most GCNs have fixed graph structures, recent work [12,35] find it beneficial to recalculate edges after each convolution operation. For instance, two distant chairs in the same indoor scene may be close at feature space since they have similar geometric properties. This information cannot be easily exploited using fixed graph, but it may be discovered if graph is updated at each feature embedding. In our dynamic graph construction, the k-NN for each node can be generated based on L_2 distance metric in current feature space. In case of multilayer graphs, k-nearest neighbor graph at each layer is constructed from node features of preceding layer. Since two distant patches in the original space may also have similar semantic features, the idea of this dynamic update is that this can captures nonlocal semantic information among the scene in a high-dimensional metric embedding.

To enlarge receptive field without loss of resolution or coverage, atrous convolution [36] has been introduced to semantic image segmentation and further extended to 3D data [35]. Our DEL adopts this idea to find dilated neighbors. Specifically, $k \times d$ nearest neighbors are searched and then k neighbors are selected by skipping every d neighbors.

Graph pooling In CNNs, pooling layers are used to scale down the size of feature maps and avoid overfitting. Recently, pooling operations are adopted to Graph Neural Networks (GNNs) and GCNs. Vinyals et al. [37] and Li et al. [38] used attention mechanism to aggregate node information in the graph. Zhang et al. [39] selected nodes after sorting them in descending based on their last features. Ying et al. [40] learned dense assignment matrix mapping nodes to a set of clusters. The TopK method proposed by [41] selects high scoring nodes based on a learnable projection vector $\mathbf{p}$, which is sparser than assignment matrix in [40]. To form a smaller graph at each U-Net layer, our method selects a subset of nodes according to TopK method [41]. Considering a graph with N nodes and their feature embedding $\mathbf{X}$, as well as adjacency matrix $\mathbf{A}$, a down-sampled graph is constructed as

$$\begin{aligned} \mathbf{s} &= \mathbf{X}\mathbf{p}/\|\mathbf{p}\| \\ i &= TopK(s, r) \\ \mathbf{X}' &= (\mathbf{X} \odot \tanh(s))_i \\ \mathbf{A}' &= \mathbf{A}_{i,i} \end{aligned} \tag{2}$$

where $TopK$ operation ranks and returns k-largest values based on scaler projection value s; $\odot$ is element-wise multiplication and tanh is activation function that rescales logistic sigmoid. The number of k is determined by a pooling ratio r. According to selected indices i, a new graph with node feature embedding $\mathbf{X}'$ and adjacency matrix $\mathbf{A}'$ is constructed. This new graph drops $(1-r)N$ nodes from the original graph.

Graph edge convolution The proposed method applies convolution-like operations on the edges connecting neighboring nodes. A single ECL consumes input graph $G^{(l)}$ that consists of nodes $V = \{v_1, v_2, \ldots, v_N\}$ with F-dimensional features for each node, where l is the layer number and N is the number of nodes. The output is a new graph $G^{(l+1)}$ in layer $(l+1)$ that consists of nodes $V' = \{v'_1, v'_2, \ldots, v'_{N'}\}$ with F'-dimensional features for each node. Generally, graph convolution on irregular data domains is expressed as neighboring aggregation. Given the feature embedding $\mathbf{X}^{(l)} =$

$\{\mathbf{x}_1^{(l)}, \mathbf{x}_2^{(l)}, \ldots, \mathbf{x}_F^{(l)}\}$ in layer l, the graph edge convolution can be calculated as

$$\mathbf{x}_i^{(l+1)} = \alpha^{(l)}{}_{j\in N(i)}(\varphi^{(l)}(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)} - \mathbf{x}_i^{(l)})) \tag{3}$$

where $\mathbf{x}_i$ is the features of v_i , $\mathbf{x}_j$ is the features of an adjacent node; φ denotes parametric non-linear function that gathers information from neighbors; α denotes a differentiable and permutation function, e.g., mean, max or sum. Instead of using the feature of neighboring nodes directly, graph edge convolution compiles difference of features between vertex v_i and all of its neighbors. Concatenating node feature $\mathbf{x}_i$ and relative feature $\mathbf{x}_i - \mathbf{x}_j$ has been proven to perform better than using node feature only [12], since this concatenation combines both the shape information and local neighboring context. Here in our implementation, φ is chosen to be a MLP with ReLU as activation function; α is chosen to be max operation to aggregate learned information and output new features as

$$\mathbf{x}_i^{(l+1)} = \max^{(l)}{}_{j\in N(i)}(\mathrm{MLP}^{(l)}(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)} - \mathbf{x}_i^{(l)})) \tag{4}$$

where MLP function has multiple fully connected layers using concatenated features as input; the number of neighboring nodes is determined by our dynamic graph construction.

3.3 DGU module

In order to extract local contextual features of point cloud, it is necessary to process graph data through multiple levels. To achieve accurate semantic segmentation, the encoder–decoder architecture is widely employed to capture multi-level features [42]. The encoder takes input data and generates a high-dimensional feature through several convolution blocks. The decoder restores from multi-layer features aggregated by encoder. Here, we use DGU module to propagate hierarchical edge features generated by DEL and ECL. Through the dynamic graph construction, this module can not only capture edge information from initial SPG but also from multi-level feature embeddings. The structure of DGU is illustrated in Fig. 3.

U-shape networks U-shape architecture [43], which is one of the classic encoder–decoder networks, excels in pixel-level prediction. Recently, Graph U-Net [41] employs U-shape design on graph data with graph pooling layer for node down-sampling.

Encoder of dynamic graph edge convolution Instead of static graph construction in [41], our method incorporates dynamic graph into hierarchical U-shape design. ECL is performed on input graph using Eq. (4). Afterward, the multi-layer encoding blocks are stacked and each subsequent layer

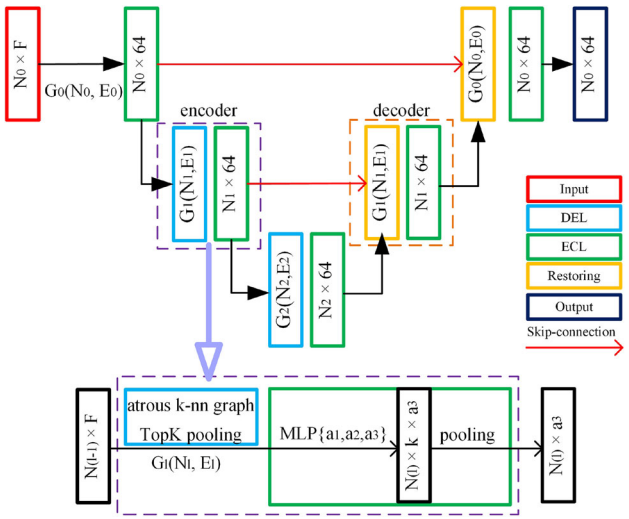


Fig. 3 An illustration of DGU module. In this 2-depths example, a ECL is first operated on input graph. One encoder contains a DEL for dynamic graph construction and node down-sampling, followed by a ECL for edge convolution on updated graph. Restoring layer brings back recorded graph structure with skip connection

operates on the output of the previous layer. Each encoding block consists of DEL and ECL. DEL is adopted to recalculate edges and thus dynamically update graph. It first constructs dynamic graph using atrous k-NN and builds a sparse adjacency matrix. Based on this adjacency matrix, a down-sampled graph is generated and recorded by graph pooling layer using Eq. (2). ECL thereafter aggregates edge features.

Decoder and skip connection The decoder stacks the same number of levels as encoder. Each decoder block has a restoring layer followed by an ECL. Restoring layer brings back the graph using recorded node structure in the corresponding DEL. The skip-connection is used between mirrored layers for feature addition. In the end, another ECL is attached to propagate hierarchical features before MLP patch prediction.

3.4 PGCNet architecture

The architecture of our PGCNet for indoor scene semantic segmentation is presented in Fig. 4. The initial SPG $G^{(l_0)}(V, E)$ is fed into DGU that carries out hierarchical edge feature aggregation. These edge features are concatenated with patch features. Subsequently, a MLP block is used to predict label of each patch. This block has three shared fully connected layers to predict label of each node, incorporating batch normalization, dropout and ReLU activation.

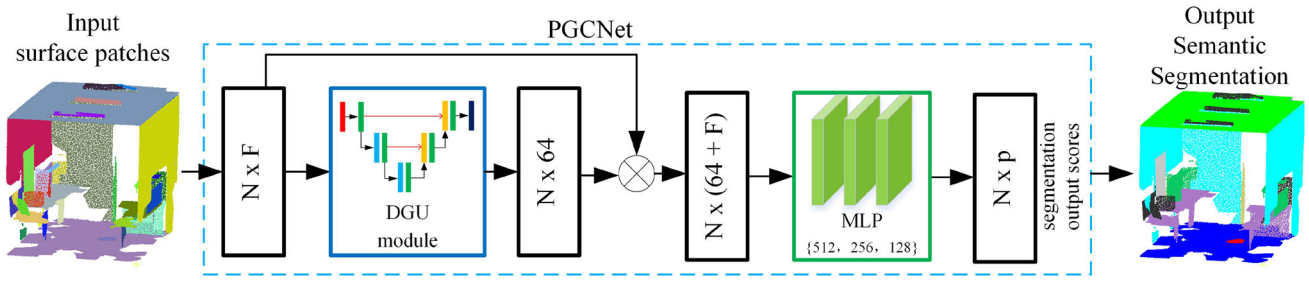


Fig. 4 The architecture of PGCNet for indoor scene semantic segmentation. Our PGCNet consumes SPG from input scene point cloud and outputs semantic segmentation. It consists of a DGU module that aggregates hierarchical edge features, and a MLP block that predicts labels for each patch. Edge features and patch features are concatenated before final prediction. N is the number of patches, F is the dimension of patch features, and the numbers in bracket are layer sizes of MLP

4 Layout-object-aware indoor scene segmentation

Indoor scene point clouds can be divided into two major categories, i.e., room layout and interior objects. Layout normally contains wall, floor, ceiling, door, window, etc. Objects include but not limited to table, chair, bookcase, etc. Based on patch features, we argue that most patches can be classified into layout or object. This classification can be regarded as a coarse segmentation, which is presumably beneficial for final rich semantic segmentation.

Layout-object classification A layout-object binary patch classification network is first trained using proposed architecture of PGCNet. This network outputs the scores of surface patches being room layout or interior objects.

Augmented semantic segmentation Regarding layout-object classification information, indoor scene semantic segmentation network is constructed by the same architecture of PGCNet. To be more specific, layout-object classification information is concatenated with the hierarchical features from DGU before MLP patch prediction block. Consequently, the output feature combines patch geometry, local context information and layout-object prediction. This concatenated feature is employed to produce final rich semantic labels for each patch.

5 Experimental results and discussions

We evaluate our proposed method first quantitative and then qualitatively on extensive experiments on public datasets.

Implementation details The proposed framework is implemented using PyTorch. The networks are trained and tested with one NVIDIA 2070 GPU. The batch size is set to 8 and the optimizer is chosen to be Adam with learning rate 0.01 initially. The learning rate is divided by 2 every 20 epochs. The depth of Dynamic Graph U-Net is 3. Inside DGU, the number k of nearest neighbors is 6 and dilation step is set to be the depth of current layer. The pooling ratio r of TopK operation is 0.5. Dropout rate is 0.2 at each layer of MLP patch prediction block.

Dataset preparation We experiment our framework on dataset S3DIS which are collected from 6 large-scale indoor areas, contains 3D point clouds with ground truth annotations. There are 271 room scenes in total. Each point is assigned to one of the semantic labels from 13 categories (wall, floor, table, chair, etc.). We also experiment our framework on ScanNet which consists of 1513 reconstructed indoor scenes with 21 classes labeling. Rather than splitting indoor scenes into blocks and sampling each block [1,11,12], our framework takes SPG from the whole indoor room as input. The thresholds for patch generation t_1, t_2, t_3 from Eq. (1) are set to 0.9, 1 cm, and 0.5 cm, respectively.

Quantitative evaluations The proposed framework is performed on ScanNet following the settings in previous approach [11]. The whole dataset are split into 1201/312 for training and testing. The RGB color information of points is removed for fair comparison, and only the XYZ information is used. The overall semantic labeling accuracy is adopted as evaluation metric.

For experiments on S3DIS, we choose the typical scene Area 5 as testing area and train our networks on the rest. Since Area 5 is not the same building as others, the indoor room scenes from this area are different to some extent. This dataset splitting is challenge but favorable for evaluating the generality of framework. Performance is evaluated by following metrics: point-wise overall accuracy (OA), class-wise mean of accuracy (mAcc), per-class intersection over union (IoU) and class-wise unweighted average of IoU of each class (mIoU). OA is defined as the proportion of correctly predicted points. For each class, the IoU is computed as

Table 2 Training time for semantic segmentation on five areas from S3DIS

Method	Scene size	Point cloud size	Data representation	Training time
DGCNN [12]	204	195 million	Sampling points	13 h
PointNet [1]			Sampling points	4 h
Ours			SPG	6 min

Table 3 Running time in seconds for semantic segmentation on typical S3DIS scene Area 5

Scene size	Point cloud size	Step	Running time
67	84 million	Patch generation	182.4
		Patch feature computation	130.4
		PGNNet-based patch labeling	0.5
		Point labeling	0.3

Table 4 Quantitative results on the Area 5 of S3DIS dataset

Method	OA	mAcc	mIoU	Ceiling	Floor	Wall	Beam	Column	window	Door	Table	Chair	Sofa	Bookcase	Board	Clutter
PointNet [1]	–	48.98	41.09	88.80	97.33	69.80	0.05	3.92	46.26	10.76	58.93	52.61	5.85	40.28	26.38	33.22
DGCNN [12]	–	–	45.97	88.13	97.41	71.40	0.11	4.88	45.50	32.29	70.99	59.11	3.50	45.33	27.89	35.43
SegCloud [21]	–	57.35	49.92	90.06	96.05	69.86	0.00	18.37	38.35	23.12	70.40	75.89	40.88	58.42	12.96	41.60
PointCNN [2]	85.91	63.86	57.26	92.31	98.24	79.41	0.00	17.60	22.77	62.09	74.39	80.59	31.67	66.67	62.05	56.74
SPGraph [25]	86.38	66.50	58.04	89.35	96.87	78.12	0.00	42.81	48.93	61.58	84.66	75.41	69.84	52.60	2.10	52.22
Patch-MLP	75.16	46.41	38.17	89.86	97.50	62.13	0.00	18.12	32.58	11.06	62.89	50.81	2.14	36.79	12.87	32.10
Ours	86.24	63.85	53.60	95.59	98.75	80.69	1.69	31.18	48.28	43.85	72.53	70.96	17.38	55.38	46.98	50.89

Table 5 Overall accuracy of ScanNet labeling

Method	OA
PointNet [1]	0.739
PointNet++ (SSG) [11]	0.833
PointNet++ (MSG) [11]	0.845
Ours	0.839

$TP/(T+P-TP)$, where TP is the number of positive points, T is the number of ground truth points of that class, and P is the number of positive points. The evaluation metrics are computed on individual sampling points. The semantic prediction for each point is generated through propagating the patch label generated by our framework.

There are five areas point cloud data for training, with million points per scene. Pointwise training [1,11,12] generally partitions the whole scene into blocks and each block is sampled with fixed number of points. In our framework, a single indoor scene is represented by a graph containing a set of surface patches. As seen from Table 2, the training time of [12] and [1] in our implementation on the same GPU is around 13 h and 4 h, respectively. In comparison, our SPG-based training time is only 6 min, which are significantly faster than pointwise training. In Table 3, we present the data statistics of running time in seconds for each step of indoor scene semantic segmentation test on Area 5 from S3DIS dataset. The main computation time is patch generation, since the whole point clouds are taken as input. The patch label prediction is only 0.5 s owing to the patch representation of indoor scene.

Quantitative results and comparisons The quantitative results of our method and comparisons with previous state-of-art methods [1,2,12,21,25] on S3DIS Area 5 are presented in Table 4. Here, we adopt our layout-object-aware PGCNet for indoor scene segmentation. Patch-MLP denotes a MLP classifier using our extracted surface patches and their features. The MLP classifier is the same one as in PGCNet. Table 4 shows that Patch-MLP can achieve similar performance as [1], especially in room layout classes such as ceilings, floor, wall, column, and door. This suggests our surface patch representation has advantage of retrieving room layout structures. The results also clearly show our PGCNet framework performs comparable segmentation with much faster training process. Notably, we perform particularly better in ceilings, floor and wall, as well as objects that is not easy to distinguished from room layouts, such as bookcase and board. This is presumably due to our patch representation and layout-object-aware augmented training. On the other hand, our framework could not perform accurate segmentation as other competitive methods

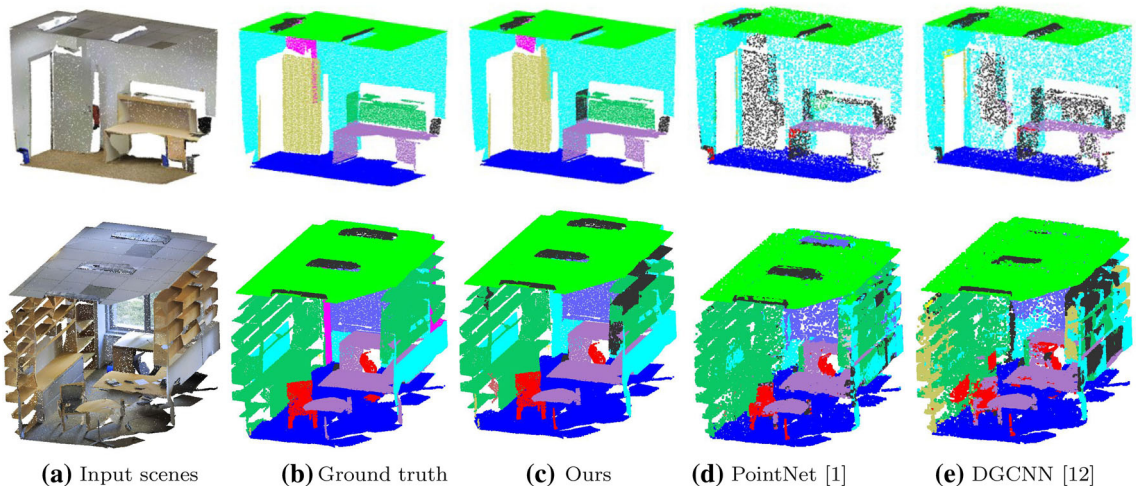


Fig. 5 Comparisons of semantic segmentation results on S3DIS scenes using PointNet [1], DGCNN [12] and our method

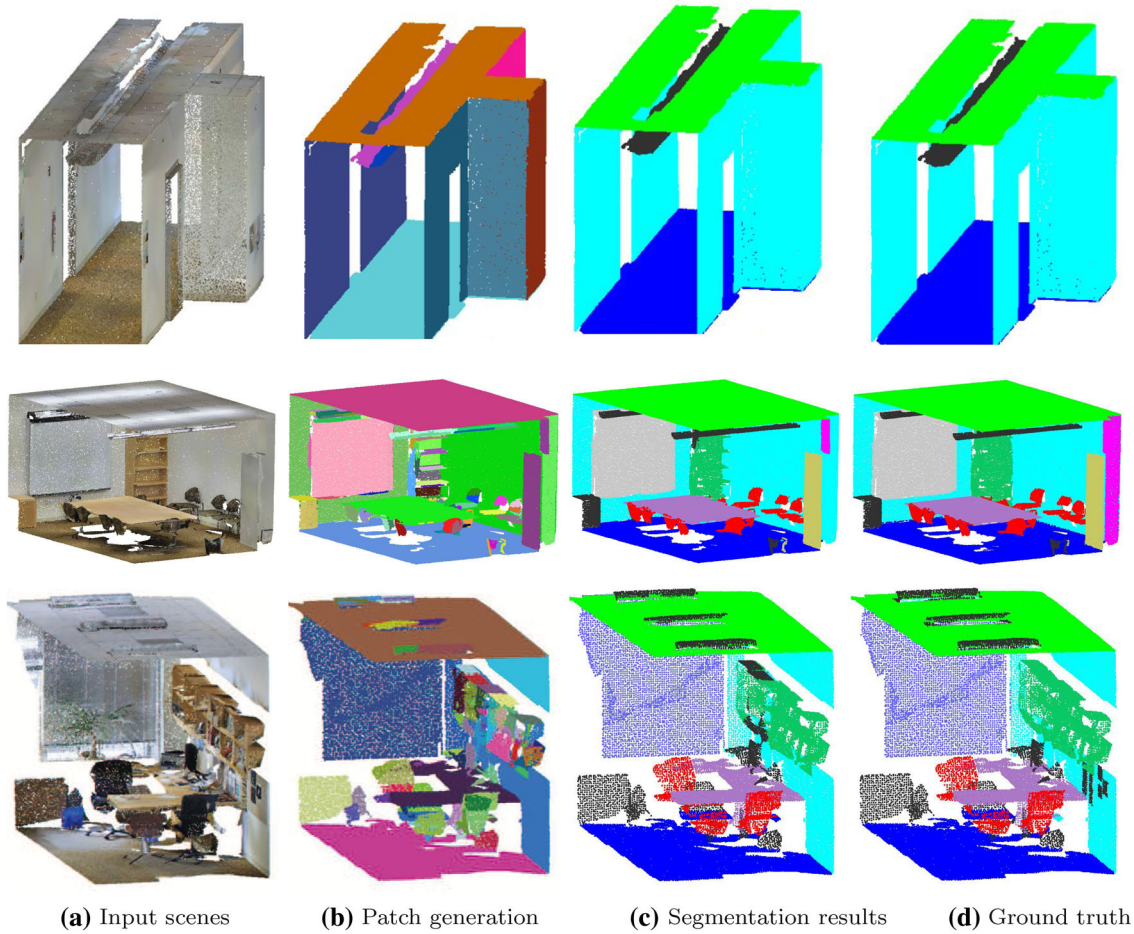


Fig. 6 Visualizations of semantic segmentation results on S3DIS scenes. Given indoor point clouds (**a**), surface patches are generated (**b**). Our PGCNet framework can achieve effective semantic segmentation performance (**c**)

for segmenting interior objects with complex shape such as sofa.

The rich labeling of ScanNet makes semantic segmentation more challenge. Using patch representation, our framework can successfully segments major layout structures and main indoor objects such as wall, floor, table, and bed. Other objects with complex shapes such as curtain and sink are difficult to distinguished from cluttered scenes. The results and comparisons of ScanNet semantic labeling accuracy is reported in Table 5. Our proposed framework outperforms [1] and achieves similar overall accuracy as [11] without pointwise multi-resolution strategy.

Qualitative results We compare our method with PointNet [1] and DGCNN [12]. Figure 5 shows visual comparisons of segmentation results on scenes from S3DIS. Notably, our method correctly segments door while others can not. Our SPG presentation carries out more smooth results than others, e.g., bookcase and table. Some chair legs and desk legs can be captured and correctly segmented, which maybe attributed to our relative features computed at neighboring edges.

More segmentation results on various types of indoor scenes are displayed in Fig. 6. Given an input indoor scene from S3DIS dataset, our method first generates surface patches as seen from Fig. 6b. Our PGCNet framework is able to correctly classify room layout including ceilings, floors, and walls as illustrated in Fig. 6c. Opened doors are almost segmented correctly in these scenes, while columns are difficult to distinguish from walls. Most indoor objects such as table and chair can be retrieved as well. The patches pasted on the wall are difficult to fully segmented, which might be part of bookcase, board or clutter.

Ablation studies To better understand the influence of each network component, we analyze them individually by removing them from PGCNet, as shown in Table 6. We conduct these studies on Area 5 of S3DIS with the same network parameters, using proposed layout-object-aware PGCNet-based method as best performance reference. NoLO model removes layout-object classification information, NoDGU model removes whole DGU module, NoDEL removes DEL and use fixed graph instead. We can see from ablation studies, removing layout-object prediction decreases the performance by nearly 2% mIoU. The DGU module and DEL accounts for 14.70% and 4.46% mIoU performance, respectively. This study suggests that proposed DGU modual significantly improves network's ability of patch feature extraction. It can also be observed that dynamic edge does play an important role in the final performance.

Table 6 Ablation studies on the S3DIS test set

Model	ΔmIoU
Best reference	0.00
NoLO	− 2.27
NoDGU	− 14.70
NoDEL	− 4.46

ΔmIoU denotes the difference in mIoU with respect to the best reference

6 Conclusions

In this work, we present a novel indoor scene segmentation framework dubbed PGCNet which uses surface patches as an efficient data representation for large-scale point clouds. Our method first extracts surface patches from indoor scene point clouds and feeds them into PGCNet. The proposed DGU module dynamically updates graph structures at each U-shape encoder to aggregate hierarchical edge features. Our network generates labels for indoor scene data, performing effective semantic segmentation results. In the future, we plan to further improve patch generation and extend PGCNet to other indoor tasks.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest. This research is supported by the National Natural Science Foundation of China under Grant No. 61972458, the Natural Science Foundation of Zhejiang Province under Grant No. LY18F020035, and the Science Foundation of Zhejiang Sci-Tech University under Grant No. 17032001-Y.

References

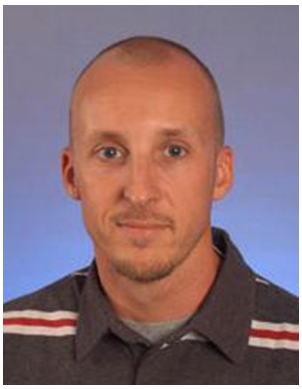
1. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)
2. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: convolution on x-transformed points. In: Advances in Neural Information Processing Systems, pp. 820–830 (2018)
3. Guo, Y., Wang, F., Xin, J.: Point-wise saliency detection on 3d point clouds via covariance descriptors. Vis. Comput. **34**(10), 1325–1338 (2018)
4. Guo, H., Zhu, D., Mordohai, P.: Correspondence estimation for non-rigid point clouds with automatic part discovery. Vis. Comput. **32**(12), 1511–1524 (2016)
5. Wirth, F., Quchl, J., Ota, J.M., Stiller, C.: Pointatme: efficient 3d point cloud labeling in virtual reality. In: IEEE Intelligent Vehicles Symposium, pp. 1693–1698 (2019)
6. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3d point cloud based object maps for household environments. Robot. Auton. Syst. **56**(11), 927–941 (2008)

7. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: IEEE International Conference on Robotics and Automation, pp. 3357–3364 (2017)
8. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J.: 3d object recognition in cluttered scenes with local surface features: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**(11), 2270–2287 (2014)
9. Maturana, D., Scherer, S.: Voxnet: a 3d convolutional neural network for real-time object recognition. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 922–928 (2015)
10. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 945–953 (2015)
11. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems, pp. 5099–5108 (2017)
12. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. **38**(5), 1–12 (2019)
13. Te, G., Hu, W., Zheng, A., Guo, Z.: Rgcnn: Regularized graph cnn for point cloud segmentation. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 746–754 (2018)
14. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of largescale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1534–1543 (2016)
15. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2432–2443 (2017)
16. Nan, L., Xie, K., Sharf, A.: A search-classify approach for cluttered indoor scene understanding. ACM Trans. Graph. **31**(6), 1–10 (2012)
17. Li, Y., Dai, A., Guibas, L., Nießner, M.: Databaseassisted object retrieval for real-time 3d reconstruction. Comput. Graph. Forum **34**, 435–446 (2015)
18. Shi, Y., Long, P., Xu, K., Huang, H., Xiong, Y.: Datadriven contextual modeling for 3d scene understanding. Comput. Graph. **55**, 55–67 (2016)
19. Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., Pajarola, R.: Object detection and classification from large-scale cluttered indoor scans. Comput. Graph. Forum **33**, 11–21 (2014)
20. Hu, S.M., Cai, J.X., Lai, Y.K.: Semantic labeling and instance segmentation of 3d point clouds using patch context analysis and multiscale processing. IEEE Trans. Vis. Comput. Graph. **26**(7), 2485–2498 (2020)
21. Tchapmi, L., Choy, C., Armeni, I., Gwak, J., Savarese, S.: Segcloud: semantic segmentation of 3d point clouds. In: Proceedings of the IEEE International Conference on 3D Vision, pp. 537–547 (2017)
22. Wang, W., Yu, R., Huang, Q., Neumann, U.: Sgpn: similarity group proposal network for 3d point cloud instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2569–2578 (2018)
23. Zhou, Y., Tuzel, O.: Voxelnet: end-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4490–4499 (2018)
24. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4421–4430 (2019)
25. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4558–4567 (2018)
26. Landrieu, L., Boussaha, M.: Point cloud oversegmentation with graph-structured deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7440–7449 (2019)
27. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: a review of methods and applications. arXiv:1812.08434 (2018)
28. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, pp. 3844–3852 (2016)
29. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv:1609.02907 (2016)
30. Peng, N., Poon, H., Quirk, C., Toutanova, K., Yih, Wt: Cross-sentence n-ary relation extraction with graph LSTMs. Trans. Assoc. Comput. Linguist. **5**(1), 101–115 (2017)
31. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv:1710.10903 (2017)
32. Shi, Y., Xu, K., Niessner, M., Rusinkiewicz, S., Funkhouser, T.: Planematch: Patch coplanarity prediction for robust rgb-d reconstruction. In: Proceedings of the European Conference on Computer Vision, pp. 750–766 (2018)
33. Huang, J., Dai, A., Guibas, L.J., Nießner, M.: 3dlite: towards commodity 3D scanning for content creation. ACM Trans. Graph. **36**(6), 203-1 (2017)
34. Mura, C., Mattausch, O., Villanueva, A.J., Gobbetti, E., Pajarola, R.: Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput. Graph. **44**, 20–32 (2014)
35. Li, G., Muller, M., Thabet, A., Ghanem, B.: Deepgcns: can gcns go as deep as cnns? In: Proceedings of the IEEE International Conference on Computer Vision, pp. 9267–9276 (2019)
36. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv:1511.07122 (2015)
37. Vinyals, O., Bengio, S., Kudlur, M.: Order matters: sequence to sequence for sets. arXiv:1511.06391 (2015)
38. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. arXiv:1511.05493 (2015)
39. Zhang, M., Cui, Z., Neumann, M., Chen, Y.: An end-to-end deep learning architecture for graph classification. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp. 4438–4445 (2018)
40. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: Advances in Neural Information Processing Systems, pp. 4800–4810 (2018)
41. Gao, H., Ji, S.: Graph u-nets. In: Proceedings of the International Conference on Machine Learning, pp. 2083–2092 (2019)
42. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
43. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241 (2015)

Yuliang Sun is a Ph.D. student in the College of Computer Science and Technology, Zhejiang University of Technology, China. He received his master degree from the University of Manchester, UK, in 2011 and received his bachelor degree from Dalian University of Technology, China, in 2010. His research interests include computer graphics, computer vision, and digital geometry processing.

Yongwei Miao received his Ph.D. degree in computer graphics from the State Key Laboratory of CAD & CG at Zhejiang University in March 2007. From February 2008 to February 2009, he worked as a visiting scholar in the University of Zürich, Switzerland. From November 2011 to May 2012, he worked as a visiting scholar in the University of Maryland, USA. From July 2015 to August 2015, he worked as a visiting professor in the University of Tokyo, Japan. Dr. Miao is currently a professor in the College of Information Science and Technology, Zhejiang Sci-Tech University, China, and also a professor in the College of Computer Science and Technology, Zhejiang University of Technology, China. His research interests include computer graphics, computer vision, digital geometry processing, visual media computing, and 3D reconstruction.

Jiazhou Chen is an associate professor in the College of Computer Science and Technology, Zhejiang University of Technology, China. Before that, he was a joint Ph.D. student between Bordeaux University, France, and Zhejiang University, China, obtaining a French doctoral diploma from Bordeaux University in July 2012, and also a Chinese doctoral diploma from Zhejiang University in December 2012. His research interests include computer graphics, image and video stylization, augmented reality, and visual media computing.



Renato Pajarola received his Dr.Sc. Techn. in computer science in 1998 from the Swiss Federal Institute of Technology (ETH) Zürich. After a postdoc in the Graphics, Visualization and Usability Center at Georgia Tech., he joined the University of California Irvine in 1999 as an assistant professor where he founded the Computer Graphics Lab. He is currently a professor in the Department of Informatics, University of Zürich, Switzerland. His research interests include computer graphics, computer vision, scientific visualization, and interactive 3D multimedia.