GECNN for Weakly Supervised Semantic Segmentation of 3D Point Clouds

Zifen HE†, Shouye ZHU†, Ying HUANG†, Nonmembers, and Yinhui ZHANG†(a), Member

SUMMARY This paper presents a novel method for weakly supervised semantic segmentation of 3D point clouds using a novel graph and edge convolutional neural network (GECNN) towards 1% and 10% point cloud with labels. Our general framework facilitates semantic segmentation by encoding both global and local scale features via a parallel graph and edge aggregation scheme. More specifically, global scale graph structure cues of point clouds are captured by a graph convolutional neural network, which is propagated from pairwise affinity representation over the whole graph established in a d-dimensional feature embedding space. We integrate local scale features derived from a dynamic edge feature aggregation convolutional neural networks that allows us to fusion both global and local cues of 3D point clouds. The proposed GECNN model is trained by using a comprehensive objective which consists of incomplete, inexact, self-supervision and smoothness constraints based on partially labeled points. The proposed approach enforces global and local consistency constraints directly on the objective losses. It inherently handles the challenges of segmenting sparse 3D point clouds with limited annotations in a large scale point cloud space. Our experiments on the ShapeNet and S3DIS benchmarks demonstrate the effectiveness of the proposed approach for efficient (within 20 epochs) learning of large scale point cloud semantics despite very limited labels.

key words: semantic segmentation, 3D point clouds, convolutional neural network, weakly supervised

1. Introduction

In addition to the ordinary RGB color information, 3D point cloud data contains intrinsic depth and geometry information of the real world. Semantic segmentation of point clouds is a fundamental problem for intelligent agents such as self-driving cars and mobile robots to understand the scene around them. Consumer level 3D sensors make it possible to capture large scale 3D point clouds and most of the recent works are focused on fully supervised semantic segmentation [1]–[3]. However, manually annotating semantic ground truth of point clouds is extremely labor intensive, which restricts rapid development of segmentation algorithms.

With this consideration, recently many research efforts are focused toward the aspect of weakly supervised semantic segmentation of point clouds. Weakly supervised approach aims at using incomplete or inexact label samples to alleviate intensive labor efforts of point level annotation of 3D data. Guinard et al. [4] propose to employ conditional random field for weakly supervised segmentation of urban scenes from 3D LiDAR point clouds. Compared with deep convolutional neural network models, the conditional random field model is usually limited by its high level abstraction ability for complex scene understanding. Mei et al. [5] propose to use semi-supervised learning for 3D LiDAR dynamic scene segmentation through depth mapping via convolutional neural networks. However, this approach could not handle 3D point clouds directly and restricted by the temporal consistency constrain imposed on semi-supervised learning.

The multi-path region mining (MPRM) [6] model leverages class activate map to explore object localization cues from weak labels that indicates categories of input spherical point samples at inexact sub-cloud level, different attention mechanisms are explored to learn a point cloud scene segmentation network by mining long-range spatial context, channel inter-dependency and global context on raw 3D data. But the limited number of pooling layers will inevitably restrict the size of receptive field thus hard for complex scene abstraction. Another line of work is based on incomplete weakly supervised segmentation [7] through ensemble learning by introducing additional constraints such as augmented Siamese network, inexact multi-instance learning and spatial smoothness loss. This approach achieves significantly higher segmentation scores even at the very limited 1% and 10% weak-supervision level.

The main motivation of our work is to solve the weakly supervised segmentation problem in an effective and efficient manner. We expect to find a way to enable efficient training within a limited number of epochs by taking into account of global graph manifold structure so as to relax localized spatial and color smoothness constraint in the ensembled objective function.

Our contribution in this work can be summarized as follows. We propose a semantic segmentation method that estimates 3D point cloud categories by encoding both global and local features via a novel graph and edge convolutional neural network architecture. Global cues of point cloud samples are incorporated by first-order approximation of graph manifold structure in order to alleviate ambiguous localized smoothness constraint imposed on isolated edge features. We demonstrate the effectiveness of the proposed weakly supervised semantic segmentation approach for efficient (within 20 epochs) learning of large scale point cloud
semantics despite very limited (1% and 10%) labels.

2. The Proposed GECNN Model

Our goal is to achieve an effective and efficient segmentation of semantic regions in large scale 3D point clouds with limited human annotation labels. We propose a novel semantic segmentation model by integrating global and local scale 3D point cloud features into an end-to-end weakly supervised convolutional neural network framework.

2.1 Problem Formulation

Consider an input point cloud data $P = \{p \in \mathbb{R}^{3}\}_{n=1}^{N}$ consisting of $N$ points, each point $p$ is located at coordinate XYZ in the 3D space. The problem of semantic segmentation is formulated as finding a labeling function that assigns a label $y_n \in Y$ for each point $p$, in which we denoted the one-hot encoded label set as $Y \in \{0, 1\}^{N \times K}$ for point cloud $P$ to be segmented into $K$ semantic categories. In the case of semantic segmentation, the goal is to segment $3$D points $\Psi_k = \{p | y_n = k\}$ from the other points $\Psi_{\neq k} = \{p | y_n \neq k\}$ so as to form a partition of $P$ such that $\Psi_k \cup \Psi_{\neq k} = P$ and $\Psi_k \cap \Psi_{\neq k} = \emptyset$ for all $k \in \{1, \ldots, K\}$.

The detailed model architecture of the proposed graph and edge convolutional neural network (GECNN) is illustrated in Fig. 1. Given the input point clouds $P$, the GECNN model computes semantic predictions through the two separated aggregation branches, namely the global feature aggregation branch (via graph feature encoding) and local feature aggregation branch (via edge feature encoding).

2.2 Global Feature Aggregation

The key building block of our global feature aggregation approach consists of three modules including d-dimension feature embedding, adjacency matrix estimation and global feature aggregation.

**d-dimension feature embedding.** To relax local-ized smoothness regularization in the constraint set, we exploit graph convolutional networks (GCN) [8] to take global graph structure into account using first-order approximation of spectral convolution. For this reason, an undirected graph $G = (V, E)$ is built with nodes $V$ corresponding to points $P \in \mathbb{R}^{N \times 3}$ and $E \in \mathbb{R}^{N \times N}$ corresponding to weights on graph edges. In this sense, the graph $G$ is a complete graph in which the pair of distinct vertices is connected by a unique edge. Given the graph $G$, we compute adjacency matrix to represent edge weights by encoding node similarity for every edges in $E$. Let $A \in \mathbb{R}^{N \times N}$ be the normalized adjacency matrix of the complete graph $G$ with self-connections, $D_{ii} = \sum_j A_{ij}$ be the diagonal entry of the corresponding degree matrix. For robust global graph feature extraction, we chose to embed 3D features of point clouds in a higher dimensional space instead of computing edge weights merely with XYZ locations. Formally, we use $V_1 = f_1(P, \Theta_1) \in \mathbb{R}^{N \times d}$ to denote embedding of point clouds $P$ into $d$-dimensional feature space derived by a two-layer convolutional neural network $f_1$ with learnable weight parameters $\Theta_1$. In particular, both layers use a $1 \times 1$ kernel and
ReLU activations. The first layer denoted as Conv1,1 has 64 output feature maps and the second layer Conv1,2 has \( d = 16 \) feature map channels. The input tensor to \( f_1 \) is with dimension \( N \times 3 \times 1 \) by extending the last channel of \( P \), and the output tensor of \( f_1 \) is with dimension \( N \times 16 \) by meanly reducing the second last channel, i.e. the XYZ channel.

**Adjacency matrix estimation.** Each entry of the adjacency matrix is evaluated by pairwise affinity \( A_{1,ij} = \|V_{ij} - V_{il}\| \) for all points \( i, j \in \{1, \ldots, N\} \), where \( \|\cdot\| \) denotes Euclidean distance operation in the \( d \)-dimensional embedding space.

**Global feature aggregation.** Given the adjacency matrix, we formulate the global feature aggregation module using approximation of spectral graph convolution to take overall graph structure into account. More formally, we adopt first-order approximation of spectral convolution

\[
g(P, A, \Theta_1) = [D_1^{-1/2}A_1D_1^{-1/2}] \times f_1(P, \Theta_1)
\]

(1) to integrate the whole graph structure into segmentation framework, where \( \times \) denotes the operator of matrix multiplication and \( D_1 \) denotes the corresponding degree matrix of \( A_1 \). The support function \( D_1^{-1/2}AD_1^{-1/2} \in \mathbb{R}^{N \times N} \) aggregates global information about feature embedding of point clouds than the localized smoothness constraint imposed by standard Laplacian regularization [10] on pairwise nodes of the graph manifold. This is potentially beneficial for reducing the effects of ambiguous segmentation of partially labeled point clouds and will become evident in the subsequent experimental sections after presenting per category semantic segmentation performances. In addition, the computational load of first-order approximation of spectral convolution scales linearly with the number of graph edges \( |E| \) thereby achieving efficient aggregation of global graph structure features even with a large number of points.

To achieve a deeper aggregation of the global features, we stack two global feature aggregation building blocks on top of each other. Similarly, given the output \( P_1 = g(P, A_1, \Theta_1) \) of the first building block, \( d \)-dimensional feature mapping of the second building block is derived by \( V_2 = f_2(P_1, \Theta_2) \) by a two-layer convolutional neural network \( f_2 \) with learnable weight parameters \( \Theta_2 \). Identically, both layers use a \( 1 \times 1 \) kernel and ReLU activations, Conv2,1 consists of 64 output feature maps and Conv2,2 consists of \( d = 16 \) feature map channels. The input tensor to \( f_2 \) is with dimension \( N \times d \times 1 \) by extending the last channel of \( P_1 \) and the dimension of the output tensor \( V_2 \) is \( N \times d \). After computing the pairwise adjacency matrix \( A_{2,ij} = \|V_{2,j} - V_{2,l}\| \), the output of the second global feature aggregation building block is \( g(P_1, A_2, \Theta_2) = [D_2^{-1/2}A_2D_2^{-1/2}] \times f_2(P_1, \Theta_2) \).

The output of the second global feature aggregation building block \( P_2 = g(P_1, A_2, \Theta_2) \) is then fed into the following pipeline to derive per category prediction of each point. For this, we first feed \( P_2 \) into a two-layer convolutional neural network \( f_3 \) with learnable weight parameters \( \Theta_3 \), both layers also use a \( 1 \times 1 \) kernel and ReLU activations. Conv3,1 consists of 40 output feature maps and Conv3,2 consists of \( K \) feature map channels. Note that for the ShapeNet dataset, the number of semantic categories is \( K = 50 \). After squeezing the second last channel of \( V_1 = f_3(P_2, \Theta_3) \), we get the final output of global feature aggregation \( \hat{p}(y|P, g, \Theta) \in \mathbb{R}^{N \times K} \).

Note that \( g \) is composed by two blocks of global feature aggregation \( g = \{g_1, g_2\} \), and the parameter \( \Theta \) is composed by the three convolutional modules \( \Theta = \{\Theta_1, \Theta_2, \Theta_3\} \).

### 2.3 Local Feature Aggregation

For local feature aggregation, our work goes along the line of works started in DGCNN [11] due to its superior performance through dynamic aggregating localized edge features on top of the \( k \)-NN algorithms. The default DGCNN stacks five local feature aggregation building blocks together, and each block consists of four modules, namely, adjacency matrix, \( k \)-nearest neighbor, edge feature and convolution (point cloud transform for the first block). The adjacency matrix \( A \) captures the same pairwise affinity as that of the global feature aggregation stage. Let \( N(P, A, k) \in \mathbb{R}^{N \times k} \) denote the set of \( k \)-nearest neighbors of point cloud \( P \) with respect to pairwise distance represented by the adjacency matrix. The edge feature \( e(P, N) \in \mathbb{R}^{N \times k \times d} \) captures the distance between each point \( P_i \) and its \( k \) neighbors \( P_j \in N_i \) in the \( d \)-dimensional feature space. The edge feature \( e \) is then fed into a single-layer convolutional neural network \( f'(e, \Gamma) \in \mathbb{R}^{N \times k \times d'} \) with learnable weight parameters \( \Gamma \). This layer employs a \( 1 \times 1 \) kernel and ReLU activation, the number of convolutional feature map output channels \( d'' \) of the corresponding four building blocks is \( 64, 64, 64 \) and 128, respectively. After taking reduced summation along the second last channel, i.e. sum over the \( k \) nearest neighbors, local feature aggregation \( f'(e, \Gamma) \) of the last four blocks are concatenated along \( d'' \) channel to combine multiscale features. The last convolutional layer extend the sum reduced feature map to 1024 channels, and then fed into a MLP with three fully connected layers, which consists of 512, 256 and \( K \) output channels. The output of the last fully connected layer produces a logistic distribution \( \hat{p}(y|P, e, \Gamma) \in \mathbb{R}^{N \times K} \) over the number of \( K \) category labels. Note that \( e \) is composed by five blocks of local edge feature aggregation, \( \Gamma \) is composed by the parameters in five convolutional layers and three fully connected layers.

### 2.4 Feature Fusion

Given the logistic predictions through global graph feature encoding branch and local edge feature encoding branch, we implement global and local feature fusion in a straightforward way through

\[
p(y|P, g, e, \Theta, \Gamma) = \hat{p}(y|P, g, \Theta) + \hat{p}(y|P, e, \Gamma)
\]

(2) Following the weakly supervised segmentation strategy in [7], we choose to define the total loss objective function \( L_{total}(\hat{y}, p(y|P, g, e, \Theta, \Gamma)) \) by composing the following four constraints based on the partially labelled ground truth \( \hat{y} \) of point cloud semantic category:
\[ L_{\text{total}} = L_1 + L_2 + L_3 + L_4 \]  

The first constraint is the incomplete supervision loss by computing softmax cross-entropy \( L_1 \) on the subset of points \( M \in \{0,1\}^N \) with ground truth labels. The second constraint is constructed in a multi-instance learning manner by computing sigmoid cross-entropy \( L_2 \) of max pooled ground truth and prediction through \( L_2 = \text{SIGM}_{\text{CE}}(\max(\bar{y}), \max(p(y))) \). The third constraint takes Siamese self-supervision into account by imposing \( L_3 \) distance divergence between the predictions of original point cloud \( P \) and transformed point cloud \( \tilde{P} \) via \( L_3 = \|p(y|P, g, e, \Theta, \Gamma) - p(y|\tilde{P}, g, e, \Theta, \Gamma)\| \). The last constraint imposes spatial and color smoothness by a manifold regularizer through \( L_4 = \frac{1}{|M|} \sum_{i=0}^{M-1} \text{tr}(p(y)^T L p(y)) \), where \( L = D - A \) denotes the Laplacian matrix of the graph.

3. Experiments

3.1 Validation on ShapeNet

We evaluate our GECNN approach for semantic segmentation on the publicly available ShapeNet [9] dataset, which is a large scale CAD model repository consists of 16881 shapes from 16 categories, each annotated with 50 semantic part categories at per-point labeling level. The ShapeNet dataset consists of 12137, 1870 and 2874 point cloud samples in train, validation and test dataset, respectively. There are a number of \( N = 2048 \) points in each sample.

**Train with 1\% labeled points.** To generate weakly supervised settings, we follow the 0.01 scheme as in the state-of-the-art weakly supervised point cloud segmentation (WSPCS) [7] method, in which only \( \frac{M}{N} = 1\% \) points with ground truth labels are randomly selected from each semantic part category of a shape model, thus there are 20 points with labels in each training sample. To evaluate the segmentation performance, we use the same standard criterion as in WSPCS, namely the average accuracy (AccAvg), average mean Intersection over Union (mIoU) for all samples (SampAvg) and all categories (CatAvg). We average the segmentation performance by repeating 5 times of random initializations of convolutional weighting parameters for both WSPCS and GECNN.

We train the proposed GECNN model on a PC with Intel Xeon 2.00 GHz CPU (15 GB RAM) and Tesla K20C GPU (5 GB RAM). Initially, we set the learning rate at 1e-3 for all experiments and batchsize at 2 in consideration of the GPU memory size. We cycled over all of the training samples for 20 epochs. In dealing with edge feature encoding, we set the number of neighbors \( k = 20 \) at the \( k\text{-NN} \) computing step. To make a fair comparison, we also run the WSPCS algorithm for 20 epochs. We train the network using the segmentation loss \( L_1 \) at the first epoch, and using the total loss \( L_{\text{total}} \) for the remaining 19 epochs.

Figure 2 shows the variation of the total loss as well as three accuracy criteria with increasing number of epochs observed in the training process. Note that the loss and accuracy criteria are evaluated on the validation dataset of ShapeNet. Compared with WSPCS, we observe that the total loss of the proposed method is relatively more stable during training and asymptotically converge to a lower state than WSPCS on the validation dataset. For validation accuracy, the proposed GECNN algorithm results in a steady and consistent performance improvement over WSPCS in terms of AccAvg, CatAvg and SampAvg criteria. It is worth noting that in the training process the WSPCS approach exhibited significantly worse accuracy than GECNN in terms of SampAvg mIoU, suffering from lack of generalization to the whole validation dataset due to its limited local edge fea-
Table 1 Performance comparison to WSPCS on test dataset of ShapeNet at 1% weak supervision level.

| Model  | AccAvg (%) | SampAvg (%) | CatAvg (%) | Memory size (MB) |
|--------|------------|-------------|------------|-----------------|
| WSPCS  | 88.20      | 72.07(±2.46)| 65.33(±0.78)| 17.6            |
| Ours   | 88.90      | 74.37(±0.64)| 68.47(±0.74)| 17.7            |

Table 2 Performance comparison to WSPCS on test dataset of ShapeNet at 10% weak supervision level.

| Model  | AccAvg (%) | SampAvg (%) | CatAvg (%) | Memory size (MB) |
|--------|------------|-------------|------------|-----------------|
| WSPCS  | 88.58      | 74.33       | 68.44      | 17.6            |
| Ours   | 89.88      | 78.56       | 72.01      | 17.7            |

Nonetheless, there are still some points get wrong labels in earphone’s segmentation.

Train with 10% labeled points. In order to demonstrate the robustness of the proposed method, we train the GECNN model at various labeling rates. The segmentation performance on test dataset of ShapeNet at 10% weak supervision level is shown in Table 2. Compared with WSPCS, the segmentation accuracy in terms of SampAvg and CatAvg increased 4.23% and 3.57% respectively. This indicates the robustness of the proposed GECNN on semantic segmentation at different weak supervision levels.

3.2 Validation on S3DIS

To demonstrate the generalization of GECNN, we verified it on the S3DIS dataset. The large point cloud dataset S3DIS consists of 6 Areas, 271 scenes, which is collected indoors. There are 4096 points in each sample, which is labelled in two specific conditions with 1% and 10% labeling rates. In this experiment, we use the segmentation results of Area 5 to evaluate the proposed method. It should be noted that there are 22 output feature maps and 13 output feature maps for Conv3_1 and Conv3_2 respectively as the number of categories is K = 13 for S3DIS dataset.

We train the proposed model and WSPCS algorithm using the same parameter settings as in Sect. 3.1. As shown in Table 3, our GECNN achieves better performance than the state-of-the-art method WSPCS. The segmentation accuracy of the proposed method outperforms WSPCS by 1.15% and 0.77% in terms of CatAvg at 1% and 10% weak supervision level, respectively. The hard example to seg-
Table 3  Performance comparison to WSPCS in terms of IoU (%) on S3DIS dataset.

| Setting | Model | AccAvg (%) | CatAvg (%) | ceil. (%) | floor (%) | wall (%) | beam (%) | col. (%) | win. (%) | door (%) | chair (%) | table (%) | book (%) | sofa (%) | board (%) | clutter (%) | Memory size (MB) |
|---------|-------|------------|------------|----------|----------|---------|---------|---------|---------|---------|----------|---------|---------|---------|---------|-------------|-----------------|
| 1%      | WSPCS | 75.85      | 42.49      | 74.43    | 93.74    | 63.05   | 0.00    | 8.34    | 42.26   | 35.39   | 58.40    | 61.97   | 7.48    | 44.26   | 31.97   | 31.03       | 11.8            |
|         | Ours  | 76.97      | 43.64      | 78.95    | 94.17    | 62.79   | 0.00    | 2.98    | 42.74   | 39.70   | 57.77    | 57.90   | 18.23   | 47.61   | 33.65   | 30.84       | 11.9            |
| 10%     | WSPCS | 77.41      | 43.10      | 73.41    | 93.82    | 65.98   | 0.00    | 6.57    | 41.89   | 31.81   | 54.85    | 63.38   | 17.95   | 47.69   | 31.02   | 31.92       | 11.8            |
|         | Ours  | 77.62      | 43.87      | 82.92    | 93.91    | 64.37   | 0.25    | 7.32    | 43.46   | 34.94   | 59.43    | 58.67   | 3.73    | 46.67   | 38.97   | 35.73       | 11.9            |

Fig. 5  Qualitative examples comparing WSPCS and GECNN for S3DIS segmentation at 1% and 10% weak supervision level.

ment is the beam dataset, whose segmentation accuracy by WSPCS in terms of IoU is 0% at both labeling rates. The segmentation accuracy of the beam dataset by the proposed GECNN method increased to 0.25% at 10% labeling rate. We attribute this to the fact of the global feature extracted by GECNN model architecture. Qualitative segmentation results of S3DIS dataset are shown in Fig. 5 and it also can be observed that the segmentation results through GECNN are better. This again demonstrates that the fusion of global and local feature is more beneficial for weakly supervised semantic segmentation of 3D point clouds than the limited edge feature.

4. Conclusion

This paper has presented a robust weakly supervised semantic segmentation approach that facilitates segmenting partly labeled 3D point clouds in an efficient way. The proposed GECNN model combines global feature through graph convolution with local feature through edge aggregation. Experiments on the ShapeNet and S3DIS benchmarks demonstrate the effectiveness of the proposed approach for efficient (within 20 epochs) learning of large scale point cloud despite very limited (1% and 10%) labels. Integrating global diffusion features into the graph convolution architecture
might be an interesting point for further investigations.

Acknowledgments

This work was supported by the National Science Foundation of China under Grants 61761024 and 62061022.

References

[1] B. Graham, M. Engelcke, and L. van der Maaten, “3D semantic segmentation with submanifold sparse convolutional networks,” CVPR, pp.9224–9232, 2018.

[2] H. Thomas, C.R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L.J. Guibas, “KPConv: Flexible and deformable convolution for point clouds,” arXiv preprint arXiv:1904.08889, 2019.

[3] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, “RandLA-Net: Efficient semantic segmentation of large-scale point clouds,” CVPR, 2020.

[4] S. Guinard and L. Landrieu, “Weakly supervised segmentation-aided classification of urban scenes from 3D LiDAR point clouds,” ISPRS Workshop, vol.XLI-1/W1, pp.151–157, 2017.

[5] J. Mei, B. Gao, D. Xu, W. Yao, X. Zhao, and H. Zhao, “Semantic segmentation of 3D LiDAR data in dynamic scene using semi-supervised learning,” IEEE Trans. Intell. Transp. Syst., vol.21, no.6, pp.2496–2509, 2020.

[6] J. Wei, G. Lin, K.-H. Yap, T.-Y. Hung, and L. Xie, “Multi-path region mining for weakly supervised 3D semantic segmentation on point clouds,” CVPR, 2020.

[7] X. Xu and G.H. Lee, “Weakly supervised semantic point cloud segmentation: Towards 10× fewer labels,” CVPR, 2020.

[8] T.N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” ICLR, 2017.

[9] L. Yi, V.G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, “A scalable active framework for region annotation in 3D shape collections,” ACM Trans. Graph., vol.35, no.6, Article No. 210, 2016.

[10] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” Journal of Machine Learning Research, vol.7, no.85, pp.2399–2434, 2006.

[11] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, and J.M. Solomon, “Dynamic graph CNN for learning on point clouds,” ACM Trans. Graph., vol.38, no.5, Article No. 146, 2019.

Zifen He received the Ph.D. degree in the direction of image processing from Kunming University of Science and Technology, Kunming, China, in 2013. She is currently an associate professor with the Department of Industrial Engineering, Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology. Her main areas of research are image processing, computer vision and 3D robot vision.

Shouye Zhu is currently a postgraduate student in Kunming University of Science and Technology, Kunming, China. His main research direction is semantic segmentation of 3D point clouds.

Ying Huang is currently a postgraduate student in Kunming University of Science and Technology, Kunming, China. Her research direction is focused on semantic segmentation of 3D point clouds.

Yinhui Zhang received the Ph.D. degree in the direction of image processing from Kunming University of Science and Technology, Kunming, China, in 2010. He is currently a professor with the Department of Mechanical Engineering, Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology. His research interests include image processing, computer vision, machine vision and 3D robot vision.