Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

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

Boundary-Aware graph Markov neural network for semiautomated object segmentation from point clouds

Huan Luo [a,b], Quan Zheng [a,b], Lina Fang [c,*], Yingya Guo [a,b], Wenzhong Guo [a,b], Cheng Wang [d], Jonathan Li [e]

[a] College of Computer Science and Big Data, Fuzhou University, Fuzhou, FJ 350108, China
[b] Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou, FJ 350108, China
[c] Academy of Digital China (Fujian), Fuzhou University, Fuzhou, FJ 350108, China
[d] Fujian Key Lab of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, FJ 361005, China
[e] Department of Geography and Environmental Management and Department of Systems Design Engineering University of Waterloo, Waterloo, ON N2L 3G1, Canada

ARTICLE INFO

Keywords:
Point Cloud
3D Object Segmentation
Boundary Constraint
Graph Neural Network
Markov Random Field

ABSTRACT

Due to the advantages of 3D point clouds over 2D optical images, the related researches on scene understanding in 3D point clouds have been increasingly attracting wide attention from academy and industry. However, many 3D scene understanding methods largely require abundant supervised information for training a data-driven model. The acquisition of such supervised information relies on manual annotations which are laborious and arduous. Therefore, to mitigate such manual efforts for annotating training samples, this paper studies a unified neural network to segment 3D objects out of point clouds interactively. Particularly, to improve the segmentation performance on the accurate object segmentation, the boundary information of 3D objects in point clouds are encoded as a boundary energy term in the Markov Random Field (MRF) model. Moreover, the MRF model with the boundary energy term is naturally integrated with the Graphical Neural Network (GNN) to obtain a compact representation for generating the boundary-preserved 3D objects. The proposed method is evaluated on two point clouds datasets obtained from different types of laser scanning systems, i.e. terrestrial laser scanning system and mobile laser scanning system. Comparative experiments show that the proposed method is superior and effective in 3D objects segmentation in different point-cloud scenarios.

1. Introduction

Rapid development of 3D laser scanning technologies has made 3D data become a hot topic in the field of computer vision (Guo et al., 2020). 3D point cloud, which can be directly and rapidly collected by a variety of laser scanning systems, has been widely applied in various areas, i.e. autonomous driving (Yue et al., 2018), high-definition (HD) maps (Ma et al. 2021), robotics (Wang et al., 2021), virtual reality (Bolkas et al., 2020), etc. In practice, compared to traditional optical images, point clouds have exhibited many advantages including accurate and real-world geometric information of 3D objects, scale-invariance of 3D objects, insensitivity to the lighting conditions, etc. As a fundamental task in computer vision, scene understanding based on point clouds has shown its great potentials and attracted worldwide attentions (Nie et al., 2021).

Lately, deep neural network has been introduced and explored in efficiently processing point clouds for scene understanding (Hu et al., 2020). To guarantee sufficient supervised information for training the neural network, manually-annotating abundant point clouds should be carefully implemented by well-trained annotators (Geiger et al., 2012). Moreover, annotators usually spend much time on distinguishing individual point in some complex scenarios, e.g., the points locating in object boundaries (Luo et al., 2018). Such manual annotation is laborious and arduous. To mitigate the huge burden of traditional manual annotation and improve the efficiency of creating supervised datasets, this paper mainly focuses on a semiautomated method, which require a few human interventions, to segmenting 3D objects from point clouds.

Several research works have begun to study the topic of semi-automated segmentation in point-cloud scenes (Golovinskiy and Funkhouser, 2009; Sedlacek and Zara, 2009; Luo et al., 2021). To segment points belonging to the foreground which are entangled with the background, a foreground-background segmentation is interactively

* Corresponding author at: Academy of Digital China (Fujian), Fuzhou University, Fuzhou, FJ 350108, China (L. Fang).
E-mail addresses: Fangln@fzu.edu.cn (L. Fang), guoyy@fzu.edu.cn (Y. Guo).

https://doi.org/10.1016/j.jag.2021.102564
Received 16 August 2021; Received in revised form 14 September 2021; Accepted 27 September 2021

implemented through searching an optimal cut by a min-cut algorithm (Golovinskiy and Funkhouser, 2009). During the foreground-background segmentation, the prior knowledge is provided by interactively and manually annotating foreground points. Sedlacek and Zara, (2009) proposed a Markov Random Field (MRF) model to interactively segment 3D objects from point-cloud scenarios by modeling the constraints including distance, continuity, point density, etc. Additionally, interactively drawing and giving the strokes or clicks on the foreground points provide the segmentation cue to distinguish the foreground and background points. Although those methods can obtain a good performance, the important information of object boundaries is ignored as a cue for a precise segmentation (Luo et al., 2021).

Nowadays, deep learning on computer vision has been increasingly attracting wide attentions (Voulodimos et al., 2018). Due to the orderless and unstructured characteristic in 3D points, standard Convolutional Neural Network (CNN), which gains popularities in processing 2D images, cannot be applied directly to represent the actual structure of point clouds. Human-designed operations, such as voxelization (Wang et al., 2017) and multi-view projection (Wei et al., 2020), are introduced to map 3D points to grids. Inevitably, potential geometric information may be discarded and ignored during the mapping procedure. Owing to the recent breakthroughs in neural network, Graph Neural Network (GNN) (Zhou et al., 2020) allows to exploit graph representation to model the orderless and unstructured point clouds directly. Such graph representation treat 3D points and neighboring relationships between 3D points as graph vertices and edges, respectively. GNN can reuse such graph structure in every layer to obtain the feature representation of point clouds. Therefore, a more robust and compact of feature representation of point clouds can theoretically be generated.

In order to effectively exploit boundary information and compact feature representation for accurate object segmentation, this paper proposes a Boundary-Aware Graph Markov Neural Network (BA-GMNN), which integrates the GNN with the boundary-aware MRF model. Specifically, to overcome the great computational burden caused by the high-density points, we over-segment the point clouds and transform them to supervoxels (Lin et al., 2018), which are assumed as basic units in the segmentation process. To describe the spatial contexts between adjacent supervoxels, a MRF model is exploited to ensure a smooth segmentation (Li, 1994). To preserve the object boundaries in the segmentation, object boundaries are treated as constraints and modeled as a potential energy term in the MRF model. To obtain a robust and compact feature representation, we introduce the GNN to extract the aggregated feature from a graph structure. Therefore, we present the summary of our main contributions as follows:

(1) We propose a new method to achieve the object segmentation in point clouds with only a few human interventions, which require the annotator to draw a bounding box loosely outlining the interested object before the segmentation. The proposed method is able to largely reduce the human efforts in creating the training datasets for 3D scene understanding.

(2) A new neural network BA-GMNN is proposed to model the problem of the semiautomated 3D object segmentation from point clouds. The BA-GMNN naturally unifies the GNN and the boundary-aware MRF model into a neural network, which effectively improves the segmentation performance.

(3) To validate the effectiveness of our proposed method, extensive experiments and comparisons are performed on two datasets, i.e. VMX450 (Luo et al., 2016) and Semantic3D (Hackel et al., 2017) datasets. In addition, the experimental results demonstrate that the proposed method achieves a satisfactory performance on the 3D object segmentation.

The reminder of our work is organized as follows. Section 2 provides detailed related studies on object segmentation. Section 3 presents the proposed methods on semiautomated segmentation of 3D object from point clouds. Section 4 presents and discusses the experimental results to demonstrate the performance of the proposed method. Section 5 concludes the entire paper.

2. Related work

In this section, we cover the three aspects of the related works, i.e., semiautomated object segmentation in point clouds, boundary extraction in point clouds, and GNNs as applied in point clouds data.

2.1. Semiautomated 3D object segmentation

As an effective way to reduce the labor cost for annotating training data, semiautomatic object segmentation has been a research hotspot. In recent years, the contour-based object segmentation methods such as PolygonRNN and PolygonRNN++ modeled the object segmentation on 2D images as a polygon prediction problem and exploited the Recurrent Neural Network (RNN) to predict the polygon contour (Castrejon et al., 2017; Acuna et al., 2018). The Curve-GCN treated the object segmentation on 2D images as a vertices regression problem where the positions of all vertices in a graph structure are simultaneously predicted by an end-to-end GNN (Ling et al., 2019). The pixels located in the predicted graph structure were denoted as the interested objects. All those methods needed to manually drag a coarse box around the interested objects for providing the supervised information. Similarly, by manually providing an object location as a prior knowledge, the min-cut-based method computes a foreground-background segmentation and finds a cut on a graphical model to solve the 3D object segmentations in point-cloud scenes (Golovinskiy and Funkhouser, 2009). By manually clicking or stroking a part of 3D points to indicate the supervision information, the Graph Cut (GC) achieves object segmentation in point-cloud scenes by interactively refining the segmentation results (Sedlacek and Zara, 2009). However, those methods did not consider the object boundaries into the segmentation framework explicitly. In our previous work (Luo et al., 2021), we built a boundary-aware MRF model to impose an object boundary constraint while segmenting 3D objects in point clouds. However, the built model still exploited the handcrafted feature descriptors and lacked of a unified framework to solve the problem of 3D object segmentation using point cloud data.

2.2. Boundary extraction

The object boundary is an important prior information to depict the real shape of the objects, which assists the accuracy improvement in object segmentations (Cheng et al., 2020; Zhang et al., 2020). Ding et al. (2019) proposed the boundary-aware feature propagation module (BFP) to obtain and propagate the local features in the isolated region for learning boundary information in optical images. Zhao et al. (2019) proposed a new neural network named BSANet where a boundary awareness module to enhance the boundary features for effectively extracting the object boundaries on images. Nowadays, more and more researches have begun to focus on how to extract object boundaries in 3D point clouds. To effectively detect object boundaries in unorganized point clouds, the mean shift algorithm (Comaniciu and Meer, 2002) was introduced to locate the centroid of local point clouds iteratively. The offset distance of every point, which was calculated based on the located centroid, was used as a measure to distinguish whether the point belongs to the boundary (Ahmed et al., 2018). Gong et al. (2021) proposed a Boundary Prediction Module (BPM) to predefine boundary points in point clouds. In order to train BPM, different types of boundary points needed to be annotated first. Then, to leverage the information gap of the to predict the boundary points, BPM exploited not only each boundary point but also its neighboring points for training, so as to focus on the information gap of the local point to classify the boundary points.

2.3. Graph neural networks applied in point clouds

Since point clouds can naturally be represented by a graph structure, GNN have been leveraged in point cloud processing. Recently, the GNN-based techniques have been increasingly developed to handle the point clouds (Yin et al., 2019). A 3D GNN was proposed to conduct RGB-D semantic segmentation by encoding the object representation with a local graph structure (Qi et al., 2017). In the graph structure, the representation of each node is initialized by the image feature vector and iteratively updated by the information passed by its neighboring nodes. In addition, the GNN was exploited to semantically segment large-scale point clouds with superpoint graphs (Landrieu and Simonovsky, 2018). In each superpoint graph, every node was defined as a superpoint, which aggregates a set of similar adjacent supervoxels, and each edge encoded the contextual information between different superpoints. Wang et al. (2019) designed a dynamic GNN to achieve multiple convolutional layers by using an EdgeConv module and achieved the superior performance on many tasks, i.e. classification and segmentation in point-cloud scenarios. In order to improve the network's ability to describe the contour of point cloud objects, Wang et al. (2019) proposed graph attention convolution (GAC), which learns the object structure by dynamically adjusting the shape of convolution kernel. In addition, a new GNN named Point-GNN was proposed to introduce the auto-registration mechanism to accomplish simultaneous detection of multiple objects, which demonstrated the feasibility of GNN on object detections in 3D point clouds (Shi and Rajkumar, 2020). Although many previous studies have proved that GNN is suitable for processing point clouds, there are few studies to introduce the prior knowledge of object boundaries into the graph representations.

3. Methods

3.1. Workflow of the proposed method

Fig. 1 outlines the workflow of our proposed method as follows: firstly, a bounding box BB_a around the interested 3D object to be segmented in a given point cloud scene is loosely and manually provided by the annotator. Then, a bigger bounding box BB_b is automatically generated to locate around the bounding box $BB_{a.}$ Here, we empirically set the length and width of bounding box BB_b two times larger than those of bounding box BB_a, thereby the 3D points located between the two bounding boxes can provide adequate prior knowledge of the background points. Next, to address the computational cost caused by the large number of points, we conduct supervoxel segmentation (Lin et al., 2018) to over-segment the points in the bounding box BB_b. Moreover, the points belonging to the object boundaries are extracted by fusing the candidate boundaries from edge detection methods (Ahmed et al., 2018) and supervoxel segmentation (Lin et al., 2018). Finally, the proposed BA-GMNN is leveraged to generate the final segmentation on the point clouds.

3.2. 3D object segmentation by BA-GMNN

To exploit the boundary information and obtain a compact feature representation for accomplishing the 3D object segmentation, a BA-GMNN method is proposed to integrate the GNN with BA-MRF. A BA-GMNN includes a BA-MRF to model the joint distribution of object labels under the constructed feature representation. As shown in Fig. 2, the proposed BA-GMNN for 3D object segmentation is optimized with the variational Expectation-Maximization (EM) framework (Yang and Ji, 2019), which alternates between a label inference procedure (M−Step) and a feature learning procedure (E-Step). Specifically, in the label inference procedure, the category labels are predicted by the BA-MRF according to the object boundaries and the feature representations of supervoxels. Here, the used feature representations of supervoxels are learnt by GNN, whose parameters are learnt in the feature learning procedure. In the feature learning procedure, the supervised information is generated from the supervoxels' category labels predicted by the BA-MRF. Therefore, in this subsection, we first present the BA-GMNN to infer the object labels in the label inference procedure (M−Step). Then, learning a GNN to generate the feature for E-Step is detailed. Finally, the EM algorithm of the whole optimization of the two steps is provided.

3.2.1. Boundary-Aware Markov Random Field

We model the problem of 3D object segmentation in a given point-

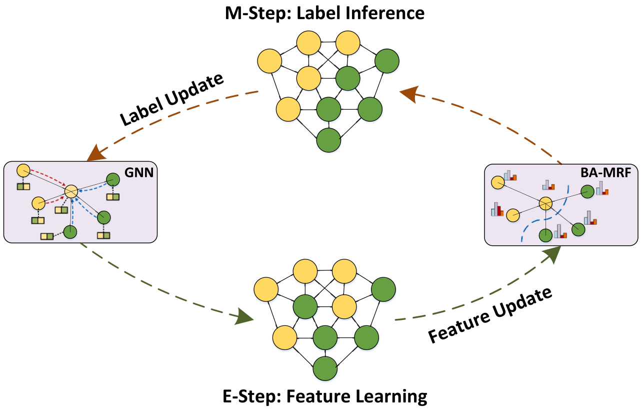

Fig. 2. The overview of BA-GMNN. There are two important steps iteratively implementing in BA-GMNN, i.e. label inference procedure (M−Step) and feature learning procedure (E-Step).

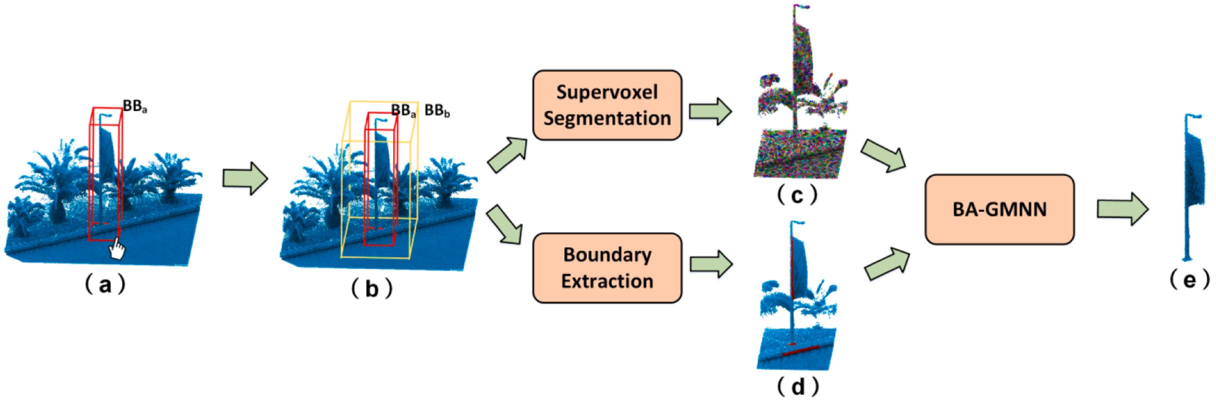

Fig. 1. Illustration of the workflow of the proposed method: (a) bounding box BBa denoted as red box which loosely outlines the interested object by the annotator; (b) bounding box BBb denoted as yellow box which is automatically generated around BBa; (c) the supervoxels obtained by the supervoxel segmentation; (d) the extracted object boundaries marked as red points; (e) the final result of 3D object segmentation.

cloud scenario c as a two-category labeling problem, whose objective is to determine the category labels $L = \{0, 1\}$ for the supervoxels $\boldsymbol{S} = \{S_1, S_2, ..., S_N\}$. Here, 1 and 0 in L represent the category label of foreground and background, respectively; N represents the total number of supervoxels extracted from the given point-cloud scenario in provided bounding box $\mathrm{BB_a}$. We formulate the energy function of BA-MRF model as follows:

$$E_{BA-MRF}(L) = \sum_{s\in S} D(L_s) + \alpha \sum_{(s_i,s_j)\in N_s} V(L_{s_i}, L_{s_j}) + \beta \sum_{(s_i,s_j)\in N_s} W(L_{s_i}, L_{s_j}) \tag{1}$$

where $D(L_s)$, $V(L_{s_i}, L_{s_j})$ and $W(L_{s_i}, L_{s_j})$ represents the data term, smooth term and boundary term, respectively. N_s represents the set of the pairwise supervoxels which are spatially-adjacent. In practice, N_s is determined on a predefined search radius R and searched through the KDTree algorithm (Li et al., 2016). Additionally, the weight factor α and β control the weight ratio of smooth term and boundary term in the energy function, respectively.

Concretely, the data term$D(L_s)$ is defined to indicate the likelihood of supervoxel S when assigning the category label L_s, as follows:

$$D(L_s) = -\log P_s\left(L_s|X^g_s\right) \tag{2}$$

where$P_s\left(L_s|X^g_s\right)$ denotes the probability of supervoxel taking category label L_s, conditioned on the feature, X^g_s. Here, X^g_s is the feature of supervoxel, which can be obtained in the feature learning procedure. To calculate the $P_s\left(L_s|X^g_s\right)$, two Gaussian Mixture Models (GMM) (Reynolds et al., 2000), i.e., GMM_b *and* GMM_f, are leveraged to represent the posterior label distribution of supervoxels for both the background and foreground. Also, the parameters of GMM_b is learnt with the supervoxels located between BB_a and BB_b, while GMM_f is learnt with the supervoxels in BB_a. Once the parameters of GMM_f and GMM_b are determined, $P_s\left(L_s|X^g_s\right)$ can be calculated as

$$P_s\left(L_s|X^g_s\right) = \sum_{i=1}^{K} w_i g_i(X^g_s; \mu_i, \Sigma_i) \tag{3}$$

$$g(X^g_s; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} exp\left[-\frac{1}{2}(X^g_s - \mu)^T \Sigma^{-1}(X^g_s - \mu)\right] \tag{4}$$

where w_i is the weight of GMM's Gaussian component i. μ_i and Σ_i are the mean vector and covariance matrix of Gaussian component i, respectively.

The smooth term $V(L_{s_i}, L_{s_j})$ encodes category dependencies between the two neighboring supervoxels S_i and S_j. We adopt the Potts model (Kohli et al., 2007) to encourage that the neighboring supervoxels with similar geometric features should be assigned the same category label. The smooth term $V(L_{s_i}, L_{s_j})$ is computed as follows:

$$V\left(L_{s_i}, L_{s_j}\right) = \begin{cases} \exp\left(-\gamma_a \|X^g_{s_i} - X^g_{s_j}\|\right), L_{s_i} \neq L_{s_j} \\ 0, L_{s_j} = L_{s_i} \end{cases} \tag{5}$$

where γ_a represents the scale factor making the smooth term comparable with other energy terms in the energy function of BA-GMNN.

The boundary term $W(L_{s_i}, L_{s_j})$ is incentive for the supervoxels positioned at segmentation boundaries to obtain all the object's boundary points. Note that, the boundary term encourages that the final segmentation should occur at the extracted boundaries. Therefore, we denote the boundary term $W(L_{s_i}, L_{s_j})$ as follows:

$$W(L_{s_i}, L_{s_j}) = \begin{cases} \exp\left(-\gamma_b\left(P^b_{s_i} + P^b_{s_j}\right)\right), L_{s_i} \neq L_{s_j} \\ 0, L_{s_j} = L_{s_i} \end{cases} \tag{6}$$

where scale factor γ_b assists in the comparability of the boundary term in Eq. (1). $P^b_{s_i}$ denotes the possibility of supervoxel S_i positioned on object boundaries. In practice, $P^b_{s_i}$ is approximately calculated as follows:

$$P^b_{s_i} = \frac{n^b_{s_i}}{n^p_{s_i}} \tag{7}$$

where $n^p_{s_i}$ and $n^b_{s_i}$denote the total points and boundary points in the supervoxel S_i, respectively. Here, $n^b_{s_i}$ can be calculated by the boundary extraction method proposed by Luo et al. (2021). Here, the object boundaries are extracted by two stages, i.e. coarse boundary generation and meaningless boundary removal. At the coarse boundary generation stage, the boundary candidates are detected by the method proposed in Ahmed et al. (2018). At the meaningless boundary removal stage, two boundary candidates provided by the supervoxels segmentation and the coarse boundary generation are naturally fused by removing the boundaries not in the intersection of those two boundary candidates. Finally, the fused boundary candidates are considered as the object boundaries which may be beneficial to accurate object segmentation.

The condition $L_{s_i} \neq L_{s_j}$ in Eq. (6) indicates that the spatially-neighboring supervoxels given with different category labels should locate at the calculated segmentation boundary. In fact, the boundary term $W(L_{s_i}, L_{s_j})$ penalizes the supervoxels at the segmentation boundary with no points belonging to object boundaries. Therefore, minimization of the boundary term benefits the object boundary preservation in the segmentation results.

Finally, due to meeting the semi-metric condition, the minimization of energy function Eq. (1) can be efficiently solved by the Graph Cuts algorithm (Boykov et al., 2001).

3.2.2. Graph neural network

Different from BA-MRF, GNN mainly focuses on learning a useful feature representation for predicting the category labels of the supervoxels. In the GNN, a graph structure $G = \{V, E\}$ is defined where the node set V, and the edge set E are denoted by supervoxels and spatially-adjacent relations between supervoxels, respectively. With the constructed graph in GNN, the category label of each supevoxel is predicted in the following way:

$$P_s\left(L_s|X^g_s\right) = Cat\left(L_s|softmax\left(\omega X^g_s\right)\right) \tag{8}$$

$$\mathrm{X}^g = g(\mathrm{X}_V, E) \tag{9}$$

where $\mathrm{X}^g \in \mathbb{R}^{|V|\times d}$ represent all supervoxels's feature representations generated by GNN, and $X^g_s \in \mathbb{R}^d$ is the GNN-generated feature representation of supervoxel S. $\omega \in \mathbb{R}^{K\times d}$ represents a matrix where d and K denote the dimension of the feature representation and the number of label categories, respectively. The function Cat stands for category distributions. X_V is the initial feature representations of all supervoxels. In practice, the X_V is calculated by applying two feature descriptors including FPFH (Rusu et al., 2009) and orientation (Munoz et al. 2009). The function g is learnt by a GNN model, GNN_ω, whereCat, E, and ω denote the input node features, the spatial relations, and the output parameters of GNN model, respectively. In addition, The GNN model is structured with 2 graph-based convolutional layers with 16 hidden units, and ReLU as the activation function (Nair and Hinton, 2010). During the GNN model training, the Adam optimizer (Kingma and Ba, 2014) minimizes the classification error under the supervision.

3.2.3. BA-GMNN optimization with EM algorithm

The proposed BA-GMNN consists of two main components: BA-MRF and GNN. Specifically, the BA-MRF models the joint probability distribution of category labels of supervoxels conditioned on their feature representations, i.e.,$P(\boldsymbol{L}|\mathrm{X}^g)$. The GNN model learns promising feature representations X^g for predicting category labels of supervoxels. However, there is no sufficient labeled supervoxels for supervising the learning procedure of the GNN. Moreover, the label inference by using BA-MRF requires the feature representationsX^g. To optimize the two components, a pseudolikelihood variational EM algorithm is proposed.

Algorithm 1 The EM algorithm for optimizing the BA-GMNN

```
Input: a graph $G = \{V, E\}$; some labeled supervoxels $(\mathrm{X}_v^l, \mathrm{L}^l)$
Output: the final labels $\mathrm{L}^u$ for unlabeled supervoxels $\mathrm{X}_v^u$.
while not converge do
    M−Step: Label Inference Procedure
    Use the $\mathrm{GNN}_\omega$ to generate the supervoxel representation, $\mathrm{X}^g$, by Eq. (9).
    Obtain the labels of unlabeled supevoxels, $\mathbf{L^u}$, with BA-MRF by minimizing the energy function (1).
    E-Step: Feature Learning Procedure
    Obtain the predicted label distribution of the unlabeled supervoxels, $P_{MRF}(\mathrm{L}^u|\mathrm{X}^g)$, by Eq. (10).
    Update the GNN, $\mathrm{GNN}_\omega$, by minimizing Eq. (11) based on $\mathrm{L}^l$ and$P_{MRF}(\mathrm{L}^u|\mathrm{X}^g)$.
end while
return $\boldsymbol{L^u}$
```

Algorithm 1 details the pseudolikelihood variational EM algorithm to optimize BA-GMNN. The EM algorithm is iteratively executed the following two steps: M−step and E-step. In M−step, we use the BA-MRF to model the local dependencies of supervoxel labels under the boundary constraint. In E-step, we update the GNN, GNN_ω, to obtain the feature representation for label prediction. Specifically, in M−step, the supervoxel feature, X^g, is generated by GNN, GNN_ω. At the first iteration, the GNN_ω is pre-trained by using the labeled supervoxels $(\mathrm{X}_v^l, \mathrm{L}^l)$. Once the X^g is obtained, the BA-MRF can be solved to generate the predicted labels for unlabeled supervoxels. Moreover, in E-step, the predicted label distribution of the unlabeled supervoxels, $P_{MRF}(\mathrm{L}^u|\mathrm{X}^g)$, can be approximated by the BA-MRF as follows:

$$P_{MRF}(\mathrm{L}^u|\mathrm{X}^g) = \frac{1}{Z(\mathrm{X}^g)} E_{BA-MRF}(\mathrm{L}^u) \tag{10}$$

where $Z(\mathrm{X}^g)$ is the normalization term. The computation of $P_{MRF}(\mathrm{L}^u|\mathrm{X}^g)$ can be effectively solved by an approximation inference method, i.e. loopy belief propagation (Murphy et al., 2013).

$$O_\omega = O_{\omega,U} + O_{\omega,L} \tag{11}$$

$$O_{\omega,U} = \sum_{X_i \in \mathrm{X}_v^u} \mathbb{E}_{P_{MRF}(L_i|X_i^g)}[P_s(L_i|X_i^g)] \tag{12}$$

$$O_{\omega,L} = \sum_{X_i \in \mathrm{X}_v^l} P_s(L_i^l|X_i^g) \tag{13}$$

where $O_{\omega,U}$ and $O_{\omega,L}$ are the objective function of GNN_ω trained by predicted label distribution of the unlabeled supervoxels and labeled supervoxels, respectively. Here, L_i^l is the category label of the annotated supervoxel, S_i, which is obtained from the prior knowledge and provided by dragging bounding box in the point-cloud scene. $O_{\omega,U}$ calculates the KL divergence between$P_{MRF}(L_i|X_i^g)$ and $P_s(L_i^l|X_i^g)$, which makes the feature representation generated from GNN are beneficial to the 3D object segmentation.

4. Experiments

4.1. Datasets

We demonstrate the capabilities of the proposed method for 3D object segmentation from point-cloud scenes, both qualitative and quantitative using evaluations performed on VMX-450 (Luo et al., 2016) and Semantic3D (Hackel et al., 2017) datasets. Particularly, the point clouds in the VMX-450 dataset were captured in Xiamen Island, Xiamen, China using a RIEGL VMX-450 mobile laser scanning (MLS) system. The MLS system, mounted on a minivan, integrates 2 RIEGL VQ-450 laser scanners, 4 high-resolution digital cameras, an inertial measurement unit (IMU), a Global Navigation Satellite System (GNSS) antenna, and a distance measurement indicator (DMI). The point density of the collected points reaches approximately 7000 points/m2 whereas the precision and accuracy are approximately 8 mm and 5 mm, respectively. The point clouds in the Semantic3D dataset were obtained in Central Europe region using a Leica ScanStation P50 and P40/P30 terrestrial laser scanning (TLS) system. Due to the high measurement resolution and long measurement range, the point density extremely changes and the occlusion largely exists. Still, the precision of the point clouds in Semantic3D dataset is within 5 mm.

Each dataset includes 50 different scenarios, which cover different types of objects such as light poles, roadblocks, billboards, traffic lights, road flags, traffic signs, trees, etc. Preservation of the object boundaries remains a major challenge for accurate 3D object segmentations. Additionally, to conveniently evaluate the proposed semiautomated method, user interventions are simulated by loosely providing a bounding box around the interested object.

4.2. Evaluation

For quantitative evaluation of the proposed method's performance for 3D object segmentation in point clouds, we used precision, recall, and F1-score (Najafi et al., 2014), which are calculated by

$$\text{precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{14}$$

$$\text{recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{15}$$

$$\text{F1-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{16}$$

where True Positives represent the correctly classified foreground points in the segmentation results. False Positives and False Negatives represent the points incorrectly classified as foreground and background points, respectively.

4.3. Experimental implementation

In our experimental setting, the supervoxel resolution R used in oversegmenting the point cloud is set at 0.125 m. During the procedure of boundary extraction, we set the resolution of the center point at 0.75 m and the search radius around the center point at 1.2 m. In the BA-MRF model, the values of weights α and β are set to (50, 200) in VMX-450 dataset, and (50, 100) in Semantic3D dataset. In addition, the scale factors γ_a and γ_b are set to 0.001 and 1. In addition, the number of neighbors of supervoxels is set to 8. Additionally, in the alternating optimization stage of the BA-GMNN for object segmentation, we train the GNN with 2 iterations and each iteration contains 200 epochs.

4.4. Results and analysis

To assess the performance of the proposed BA-GMNN in 3D object segmentation, both qualitative and quantitative evaluations were performed on all the datasets. As shown in Figs. 3 and 4, objects in different scenarios are accurately segmented, which demonstrate that the proposed BA-GMNN can accurately segment 3D objects from point-cloud scenes in VMX-450 and Semantic3D datasets. Additionally, the object boundaries were well preserved in the segmentation results, which exhibits the effectiveness of the boundary constraints modeled in the proposed BA-GMNN. Table 1 shows the overall quantitative results of our proposed method on two datasets. Moreover, the quantitative segmentation results of different objects are recorded in Table 2. As shown in Table 1, the proposed BA-GMNN achieves precision, recall, and F1-score of (0.985, 0.939, 0.959) and (0.936, 0.953, 0.940), on the two datasets, respectively. The quantitative results demonstrate the feasibility of our proposed method on 3D object segmentation of point clouds. In addition, the proposed BA-GMNN shows superior performance on the VMX-450 dataset compared to Semantic3D dataset. This is because the point density of the point clouds collected by the TLS system

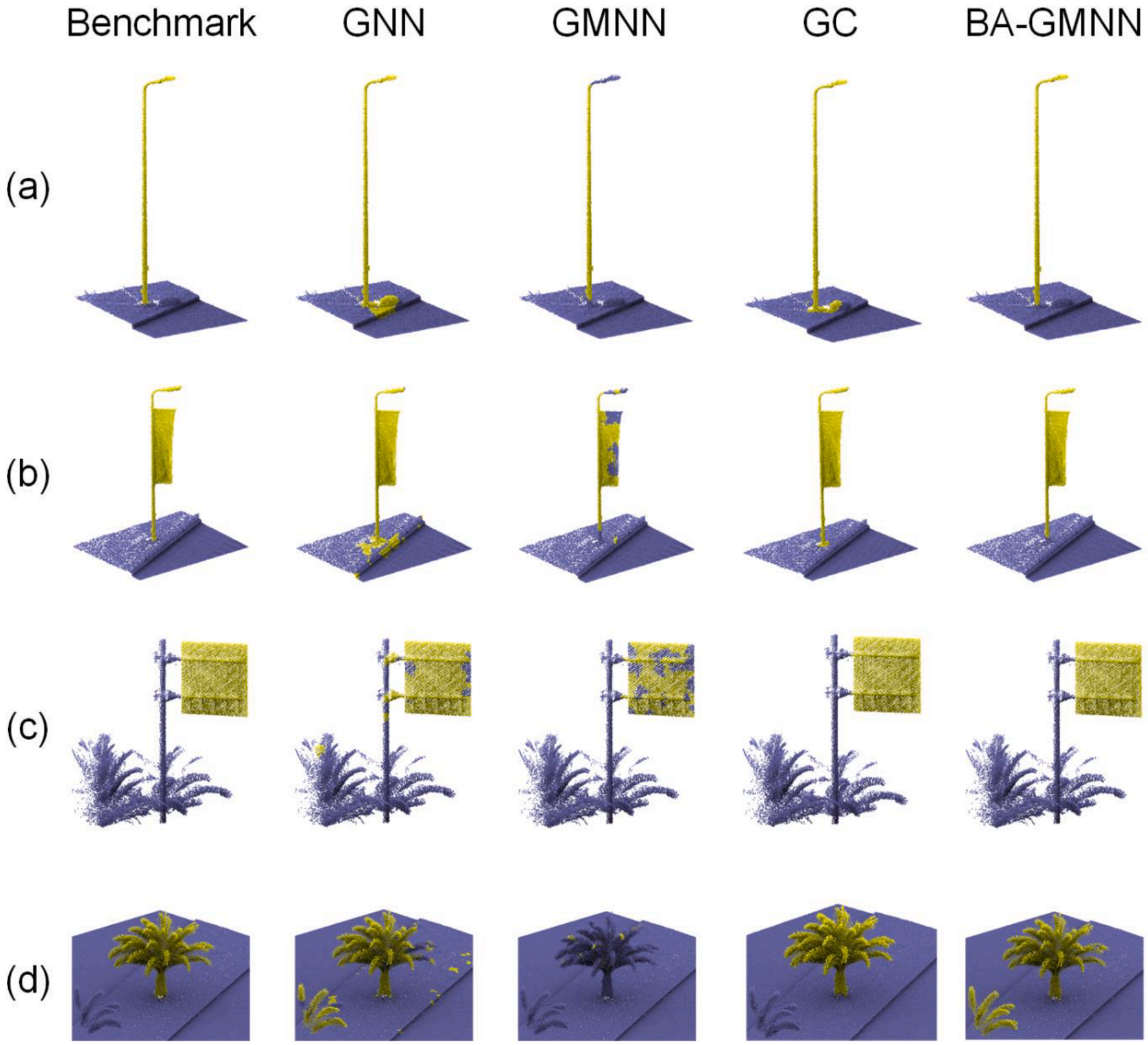

Fig. 3. Comparative segmentation results of different scenarios on the VMX-450 dataset.

changes seriously, which affects the accuracy of boundary extraction and the consistency of feature representation.

In order to exhibit the superior performance on the 3D object segmentation of point clouds, we compared the proposed BA-GMNN with three methods: (1) the GNN-based method (GNN), which treats the object segmentation as a semi-supervised problem and exploits graph neural network as the classification model. (2) the GMNN-based method (GMNN) (Qu et al., 2019), which also treats the object segmentation as a semi-supervised problem and combines the graph neural network and MRF model as its the classification model. (3) the Graph Cut-based method (GC) (Luo et al., 2016). From the comparative results exhibited in Figs. 3 and 4, we can see that other than the proposed BA-GMNN, other methods cannot effectively preserve the object boundaries during the segmentation. The GMNN-based method has a higher precision than the GNN-based method. This is because the feature representation learnt in GMNN is beneficial to the segmentation task. The GMNN-based method has a lower recall. The reason is that lacking of adequate training samples causes misclassification where true positives are wrongly classified as false negatives. By introducing boundary constraints as the prior knowledge, the proposed BA-GMNN method can introduce more true positives into the procedure of model training. In addition, as shown in Fig. 3(d), the BA-GMNN method segment all the leaves of iron trees, even in cases where the iron trees are not in the bounding box provided by the annotator. This is because the BA-GMNN method treats segmentation process as a two-phase classification problem which encourage similar supervoxels in feature space to be assigned with the same category label.

Fig. 5 gives two failure cases of our proposed method on semi-automated segmentation of 3D objects. As the first case shown in Fig. 5 (a) and (b), there are many local components with similar feature description in the background and foreground objects, i.e. the pole and the tree trunk, and the flag on the light pole and tree leaves. Our proposed BA-GMNN may obtain an inaccurate segmentation in such scenario. This is because the BA-GMNN method treats the object segmentation as a semi-supervised classification problem. Too many similar samples in different categories may influence the accuracy of the segmentation. As the second case shown in Fig. 5 (c) and (d), it is difficult to extract the boundary points in the bush segmentation. This is because too many clutter points lead to the unclear boundaries which cannot help our proposed method to separate the foreground from the background. Therefore, as shown in Table 2, the segmentation quantization result of bushes is not as good as other objects.

Our proposed method was evaluated on a workstation which is equipped with eight Intel Xeon E5-2620 processors, a memory of 125 GB and a Tesla P100 GPU, and is running at Ubuntu 16.04. We recorded the execution time of each stage in our proposed method implemented on a single thread. Specifically, the calculation time of supervoxels segmentation and feature extraction for VMX450 and Semantic3D data sets is

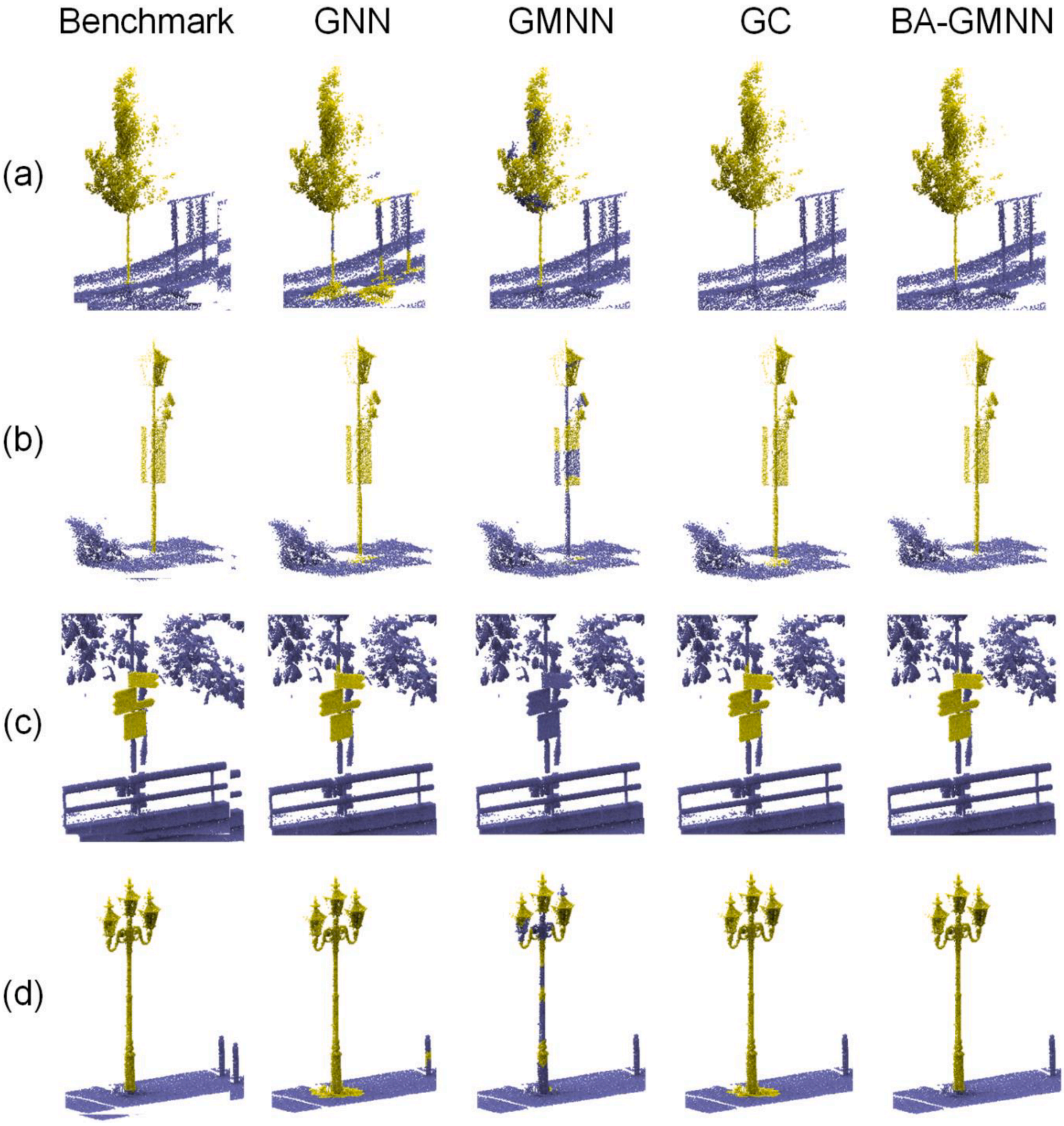

Fig. 4. Comparative segmentation results of different scenarios on the Semantic3D dataset.

Table 1
Comparison results among our method and other semiautomated segmentation methods tested on the VXM-450 and Semantic3D datasets.

	VXM-450			Semantic3D		
	Precision	Recall	F_1-score	Precision	Recall	F_1-score
GNN	0.747	0.853	0.786	0.851	0.930	0.883
GMNN	0.835	0.524	0.607	**0.963**	0.609	0.725
GC	**0.996**	0.895	0.930	0.894	**0.959**	0.918
BA-GMNN (ours)	0.985	**0.939**	**0.959**	0.936	0.953	**0.940**

0.27 h and 0.14 h, respectively. The boundary extraction for two datasets takes 0.039 h and 0.037 h, respectively. The BA-GMNN optimization for two datasets takes 0.91 h and 0.48 h, respectively.

Table 2
Experimental results of different objects by our method on the VXM-450 and Semantic3D datasets.

	VXM-450			Semantic3D		
	Precision	Recall	F_1-score	Precision	Recall	F_1-score
RoadBlock	0.971	0.934	0.953	0.981	0.919	0.948
TrafficSign	0.943	0.999	0.969	0.968	0.999	0.984
Billboard	0.995	0.913	0.948	0.968	0.981	0.974
Tree	0.959	0.994	0.976	0.998	0.997	0.998
LightPole	0.960	0.961	0.958	0.987	0.994	0.990
Bushes	0.973	0.712	0.784	0.789	0.998	0.880
Others	0.999	0.979	0.989	0.886	0.958	0.915

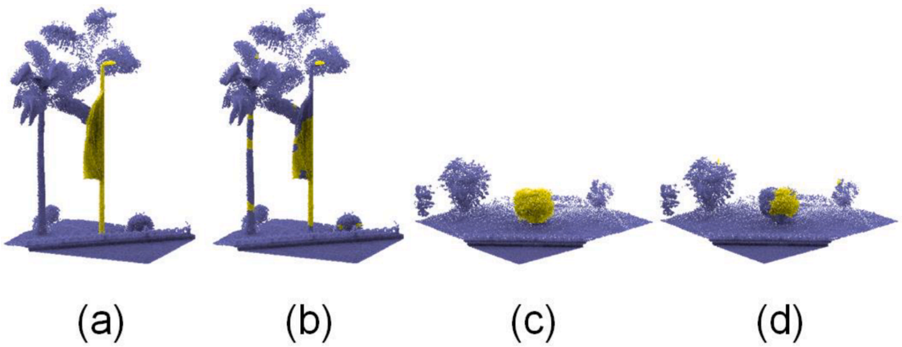

Fig. 5. Examples of failure cases of our proposed method: (a) and (c) are benchmark; (b) and (d) are segmentation results of BA-GMNN.

5. Discussion

To examine the impact of the boundary term β, on the performance of 3D object segmentation, the evaluations for the BA-GMNN on VMX450 and Semantic3D datasets are performed on the following configurations: 0, 100, 200, 300, 400, and 500. As shown in Fig. 6, as the weight of boundary term increases from 0 to 200, the value of F1-score increases gradually. This is because the boundary constraints imposed on the object segmentation are beneficial to improve the accuracy of segmentation. The peak value of F1-score is reached at 200 and remains stable between 200 and 400. Once the weight of the boundary term is larger than 400, the value of F1-score gradually decreases. This is because too larger weight of the boundary term may influence the function of the smooth term and the data term in the energy function of BA-GMNN. Similarly, as shown in Fig. 7, when the weight of boundary term ranges from 0 to 100, the performance of segmentation increases rapidly and reaches the peak at 100. When the weight of boundary term increases from 100 to 400, the value of F1-score almost remains stable. Once the weight of the boundary term is larger than 400, the performance of segmentation decreases rapidly. Therefore, for segmenting 3D objects in the VMX450 and Semantic3D datasets, we set the boundary energy weights in BA-GMNN to 200 and 100, respectively.

6. Conclusions

Our paper proposes a new method which integrates GNN and BA-MRF for effective semiautomated object segmentation from 3D point clouds. The proposed method can potentially reduce the manual annotation task of point clouds for 3D scene understanding. In order to consider neighboring contexts and boundary constraints, we propose to design a boundary term into a MRF model. Moreover, to construct a useful feature representation for predicting category labels, although there is no adequate labeled training samples, a GNN is introduced and trained by a pseudolikelihood variational EM algorithm. Extensive evaluations on two datasets collected by TLS and MLS systems demonstrates that the proposed method achieves the F1-score at 0.959 and 0.940, respectively. Related studies also show that the proposed method outperforms other methods in boundary preservation as observed in the segmentation results. In future research, it is very promising to obtain a satisfactory segmentation in the scenario where object boundaries cannot be accurately extracted. In addition, providing annotators with a more convenient approach to complete the interaction in the segmentation procedure may be another possible research direction.

This study was financially supported by National Natural Science Foundation of China (Grant No. 61801121, No. U1705262 and No. 62001175), Natural Science Foundation of Fujian Province (No. 2018J07005 and No. 2019J05034).

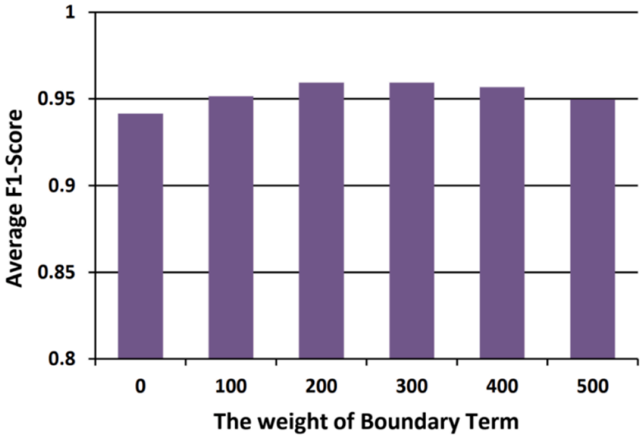

Fig. 6. Influence of the weight of boundary term on the performance of the 3D object segmentation in VMX-450 Dataset.

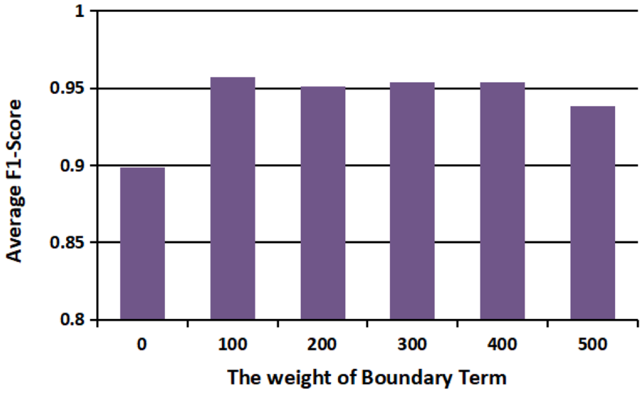

Fig. 7. Influence of the weight of boundary term on the performance of the 3D object segmentation in Semantic3D Dataset.

CRediT authorship contribution statement

Huan Luo: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing, Funding acquisition. **Quan Zheng:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Lina Fang:** Investigation, Formal analysis, Visualization, Resources, Validation, Supervision. **Yingya Guo:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Project administration, Funding acquisition. **Wenzhong Guo:** Resources, Supervision. **Cheng Wang:** Resources, Supervision. **Jonathan Li:** Validation, Resources, Supervision.

Declaration of Competing Interest

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

References

Acuna, D., Ling, H., Kar, A., Fidler, S., 2018. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 859–868.

Ahmed, S.M., Tan, Y.Z., Chew, C.M., Al Mamun, A., Wong, F.S., 2018. Edge and Corner Detection for Unorganized 3D Point Clouds with Application to Robotic Welding. In: In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 7350–7355.

Bolkas, D., Chiampi, J., Chapman, J., Pavill, V.F., 2020. Creating a virtual reality environment with a fusion of suas and TLS point-clouds. International journal of image and data fusion. 11 (2), 136–161.

Boykov, Yuri, Veksler, Olga, Zabih, Ramin, 2001. Fast approximate energy minimization via graph cuts. IEEE Transactions on pattern analysis and machine intelligence 23 (11), 1222–1239.

Castrejon, L., Kundu, K., Urtasun, R., Fidler, S., 2017. Annotating object instances with a polygon-rnn. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5230–5238.

Cheng, T., Wang, X., Huang, L., Liu, W., 2020. Boundary-preserving mask r-cnn. In: In: European conference on computer vision. Springer, pp. 660–676.

Comaniciu, D., Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence. 24 (5), 603–619.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In: In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp. 3354–3361.

Golovinskiy, A., Funkhouser, T., 2009. Min-cut based segmentation of point clouds. In: In: 2009 IEEE 12th International Conference on Computer Vision Workshops. IEEE, pp. 39–46.

Gong, J., Xu, J., Tan, X., Zhou, J., Qu, Y., Xie, Y., Ma, L., 2021. Boundary-aware geometric encoding for semantic segmentation of point clouds. arXiv preprint arXiv: 2101.02381.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2020. Deep Learning for 3D Point Clouds: A Survey. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 1 1.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J, D., Schindler, K., Pollefeys, M., 2017. Semantic3d.net: a new large-scale point cloud classification benchmark. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 91–98.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., 2020. Randla-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11108–11117.

Kingma, D., Ba, J., 2014. Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations.

Kohli, P., Kumar, M.P., Torr, P., 2007. P3 & beyond: Solving energies with higher order cliques. In: In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1–8.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4558–4567.

Li, S.Z., 1994. Markov random field models in computer vision. In: In: European conference on computer vision. Springer, pp. 361–370.

Li, S., Wang, J., Liang, Z., Lian, S., 2016. Tree point clouds registration using an improved icp algorithm based on kd-tree. In: In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp. 4545–4548.

Lin, Y., Wang, C., Zhai, D., Li, W., Li, J., 2018. Toward better boundary preserved supervoxel segmentation for 3d point clouds. ISPRS journal of photogrammetry and remote sensing. 143, 39–47.

Ling, H., Gao, J., Kar, A., Chen, W., Fidler, S., 2019. Fast interactive object annotation with curve-gcn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5257–5266.

Luo, H., Wang, C., Wen, C., Cai, Z., Chen, Z., Wang, H., Yu, Y., Li, J., 2016. Patch-based semantic labeling of road scene using colorized mobile lidar point clouds. IEEE Transactions on Intelligent Transportation Systems. 17 (5), 1286–1297.

Luo, H., Wang, C., Wen, C., Chen, Z., Zai, D., Yu, Y., Li, J., 2018. Semantic labeling of mobile lidar point clouds via active learning and higher order mrf. IEEE Transactions on Geoscience and Remote Sensing. 56 (7), 3631–3644.

Luo, H., Zheng, Q., Wang, C., Guo, W., 2021. Boundary-aware and semiautomatic segmentation of 3-d object in point clouds. IEEE Geoscience and Remote Sensing Letters. 18 (5), 910–914.

Ma, L., Li, Y., Li, J., Yu, Y., Junior, J.M., Goncalves, W.N., Chapman, M.A., 2021. Capsule-based networks for road marking extraction and classification from mobile LiDAR point clouds. IEEE Transactions on Intelligent Transportation Systems. 22 (4), 1981–1995.

Munoz, D., Bagnell, J.A., Vandapel, N., Hebert, M., 2009. Contextual classification with functional max-margin markov networks. In: In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 975–982.

Murphy, K., Weiss, Y., Jordan, M. I., 2013. Loopy belief propagation for approximate inference: An empirical study. arXiv preprint arXiv:1301.6725.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. In: 2010 27th International Conference on Machine Learning (ICML).

Najafi, M., Namin, S.T., Salzmann, M., Petersson, L., 2014. Non-associative higher-order markov networks for point cloud classification. In: In: European Conference on Computer Vision. Springer, pp. 500–515.

Nie, Y., Hou, J., Han, X., Nießner, M., 2021. RfD-Net: Point Scene Understanding by Semantic Instance Reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4608–4618.

Qi, X., Liao, R., Jia, J., Fidler, S., Urtasun, R., 2017. 3d graph neural networks for rgbd semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5199–5208.

Qu, M., Bengio, Y., Tang, J., 2019. Gmnn: Graph markov neural networks. In: In: International conference on machine learning, pp. 5241–5250.

Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker verification using adapted gaussian mixture models. Digital signal processing. 10 (1-3), 19–41.

Rusu, Radu Bogdan, Blodow, Nico, Beetz, Michael, 2009. 2009 IEEE international conference on robotics and automation. IEEE, pp. 3212–3217.

Sedlacek, D., Zara, J., 2009. Graph Cut Based Point-Cloud Segmentation for Polygonal Reconstruction. In: In: International Symposium on Visual Computing. Springer, pp. 218–227.

Shi, W., Rajkumar, R., 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1711–1719.

Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E., 2017. Deep learning for computer vision: A brief review. Computational intelligence and neuroscience, 2018.

Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J., 2019a. Graph attention convolution for point cloud semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10296–10305.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., Tong, X., 2017. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM Transactions On Graphics. 36 (4), 1–11.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019b. Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics. 38 (5), 1–12.

Wang, X., Mizukami, Y., Tada, M., Matsuno, F., 2021. Navigation of a mobile robot in a dynamic environment using a point cloud map. Artificial Life and Robotics. 26 (1), 10–20.

Wei, X., Yu, R., Sun, J., 2020. View-gcn: View-based graph convolutional network for 3d shape analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1850–1859.

Yang, L., Ji, H., 2019. A variational em framework with adaptive edge selection for blind motion deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10167–10176.

Yin, B., Chadha, A., Abbas, A., Bourtsoulatze, E., Andreopoulos, Y., 2019. Graph-based object classification for neuromorphic vision sensing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1711–1719.

Yue, X., Wu, B., Seshia, S.A., Keutzer, K., Sangiovanni-Vincentelli, A.L., 2018. A LiDAR Point Cloud Generator: from a Virtual World to Autonomous Driving. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, pp. 458–464.

Zhang, Q., Shi, Y., Zhang, X., 2020. Attention and boundary guided salient object detection. Pattern Recognition. 107, 107484. https://doi.org/10.1016/j.patcog.2020.107484.

Zhao, Y., Li, J., Zhang, Y., Tian, Y., 2019. Multi-class Part Parsing with Joint Boundary-Semantic Awareness. In: Proceedings of the IEEE International Conference on Computer Vision(ICCV), pp. 9177–9186.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M., 2020. Graph neural networks: A review of methods and applications. AI Open. 1, 57–81.